UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA

PROYECTO DE GRADO

TITULO: SOFTWARE DE SEGUIMIENTO Y MONITOREO DE EVENTOS CULTURALES
CASO: Dirección General de Cultura – G.A.M.C

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: PETER PAUL CASTILLO VALENCIA
TUTOR METODOLOGICO: Lic. GROVER A. RODRIGUES RAMIREZ
ASESOR: M.Sc. FRANZ CUEVAS QUIROZ

LA PAZ – BOLIVIA

2015

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA

A mi padre José a quien amo, quien es mi ejemplo de vida.

A mi hija Anabel a quien amo y me da las fuerzas para salir adelante y le da sentido a mi vida.

A toda mi familia y amigos que estuvieron en cada paso que daba hacia la culminación de mis estudios.

AGRADECIMIENTOS

Quiero agradecer a Dios. Por poner en mi camino grades personas como mi familia y amigos con los cuales siempre podré contar con ellos como estuvieron en los momentos felices y más aún en los momentos difíciles de mi vida.

A mi familia que sin su apoyo en momentos críticos quizás no hubiese salido delante, y que gracias a los consejos de mi padre y de mis hermanos pude seguir adelante en este proceso.

A mi docente tutor Lic. Grover Rodríguez, por el apoyo y colaboración que me brindó para que pudiera concluir éste proyecto.

Agradezco al M Sc. Franz Cuevas Quiroz , mi docente asesor, por todo el tiempo que me dio para revisar éste trabajo. Su orientación y sugerencias me sirvieron mucho durante el tiempo que me llevo realizar el presente proyecto de grado.

A mi padre José por haberme ayudado y apoyado en todo lo que pudo, y los concejos que recibí de el en el transcurso de mi carrera.

Agradezco a todos los docentes con los cuales curse materias ya que ellos me enseñaron la base para poder potenciar mis conocimientos de la profesión.

Gracias a mis amigos que me apoyaron a lo largo de mi vida universitaria, y que de mi parte siempre tendrán a un amigo incondicional.

RESUMEN

La Dirección General de Cultura del Gobierno Autónomo Municipal de Copacabana G.A.M.C., ejecuta programas con el objeto de proteger, fomentar y promocionar el dialogo intercultural y desarrollo cultural y creativo, así como defender el patrimonio cultural de este municipio. Lo cual a influido en el crecimiento turístico. Los procesos para el seguimiento de estos programas son manuales provocando retardo en el procesamiento de información el cual no permite una adecuada toma de decisiones.

El presente trabajo, trata de la construcción de un Software orientado a la Web para el seguimiento y monitoreo de eventos culturales de la Dirección General de Cultural del G.A.M.C., utilizando metodología de desarrollo XP(Programación Extrema) como marco de trabajo. Este proyecto a sido desarrollado con el objetivo de brindar una herramienta para agilizar el proceso de recolección de información sobre los procesos de los eventos culturales que realiza esta institución para brindar información oportuna para apoyo en la toma de decisiones.

El presente proyecto se divide en 7 capítulos que a continuación se los describe:

Cáp. I Comprende las generalidades del proyecto de grado.

Cáp. II Se explica todos los conceptos que serán utilizados en el desarrollo del proyecto.

Cáp. III Comprende la Planeación, análisis, diseño e implantación del proyecto.

Cáp. IV Comprende la calidad del software, que permite evaluar el sistema .

Cáp. V En este capítulo se encuentra costos y beneficios.

Cáp. VI En este capítulo se describe elementos de seguridad.

Cáp. VII En este capítulo se encuentra conclusiones y recomendaciones.

El Software ha sido implementado en la institución cumpliendo con las expectativas y requerimientos que se plantearon al inicio de este proyecto y considerando los factores de calidad definidos en la norma ISO 9126.

ABSTRACT

The Directorate General of Culture of the Autonomous Municipal Government of Copacabana GAMC, runs programs in order to protect, promote and promote intercultural dialogue and cultural and creative development and defend the cultural heritage of this town. Which influenced tourism to growth. Processes for monitoring these programs are causing delay in manual processing of information which does not allow proper decision making.

This work is building a Web-oriented software for tracking and monitoring of cultural events of the Directorate General of Cultural of GAMC, development methodology using XP (Extreme Programming) as a framework. This project was developed with the aim of providing a tool to streamline the process of gathering information on the processes of cultural events organized by this institution to provide timely information to support decision-making.

This project is divided into 7 chapters which are described below:

Chap. I includes an overview of the graduation project.

Chap. II all the concepts that will be used in development is explained project.

Chap. III involves planning, analysis, design and implementation of the project.

Chap. IV includes software quality, which evaluates the system.

Chap. V This chapter is costs and benefits

Chap. VI This chapter describes security elements.

Chap. VII This chapter is conclusions and recommendations.

The software has been implemented in the institution meeting the expectations and requirements that were raised at the beginning of this project and considering the quality factors defined in the ISO 9126 standard.

INDICE

CAPITULO I **13**

MARCO REFERENCIAL **13**

1.1 INTRODUCCION 13
1.2 PROBLEMA 14
1.2.1 ANTECEDENTES *14*
1.2.2 PLANTEAMIENTO DEL PROBLEMA *15*
1.2.3 FORMULACIÓN DEL PROBLEMA *15*
1.3 OBJETIVOS 15
1.3.1 OBJETIVO GENERAL *15*
1.3.2 OBJETIVOS ESPECÍFICOS *15*
1.4 JUSTIFICACION 16
1.5 LIMITES Y ALCANCES 16
1.6 APORTES 17

CAPITULO II **18**

MARCO TEORICO **18**

2.1 MARCO INSTITUCIONAL 18
2.1.2 VISION *18*
2.1.3 ORGANIGRAMA *19*
2.1.4 ASPECTOS HISTORICOS Y CULTURALES *20*
2.2 DESCRIPCION DE LA DIRECCION DE CULTURA 20
2.2.1 FUNCIONES Y ATRIBUCIONES ESPECÍFICAS *20*
2.3 MONITOREO 21
2.3.1 EFECTOS E IMPACTOS *21*
2.3.3 INDICADORES *22*
2.3.4 EVALUACIÓN *22*
2.4 METODOLOGIA DE DESARROLLO DEL SISTEMA 23
2.4.1 METODOLOGIA DE DESARROLLO XP (eXtreme Programming) *23*
2.4.2 PLANEACION *27*
2.4.3 DISEÑO *30*
2.4.4 CODIFICACIÓN *32*
2.4.5 PRUEBAS *33*
2.6 WEBML (WEB MODELING LANGUAJE) 35
2.6.1 MODELO DE DATOS *36*
2.6.2 MODELO DE HIPERTEXTO *37*
2.7 TECNOLOGIA DE SOFTWARE 38
2.7.1 APACHE *38*
2.7.2 PHP *38*
2.7.3 MySQL *39*
2.7.4 AJAX *40*
2.8 CALIDAD DE SOFTWARE 41
2.8.1 ISO 9126 ..... *41*

CAPITULO III **49**

MARCO APLICATIVO **49**

3.1 INTERRELACION DE METODOLOGIAS 49
3.2 FASE DE PLANIFICACION 50
3.2.1 HISTORIAS DE USUARIO *50*
3.2.2 TASK CART (TARJETA DE TAREA) *55*
3.2.3 PLAN DE ENTREGAS *63*
3.2.4 VELOCIDAD DEL PROYECTO *64*
3.2.5 PLANIFICACION DE ITERACIONES *65*
3.2.6 METAFORA DEL SISTEMA *65*
3.3 DISEÑO 66
3.3.1 TARJETAS CRC *66*
3.3.2 MODELO DE DATOS *69*
3.3.3 PRIMERA ITERACIÓN *70*
3.4 DESARROLLO 81
3.4.1 DISEÑO SIMPLE *81*
3.4.1 PANTALLAS MUERTAS *81*
3.5 PRUEBAS 86
3.5.1 PRUEBAS DE ACEPTACIÓN *86*

CAPITULO IV **88**

MÉTRICAS DE CALIDAD **88**

4.1 ESPECIFICACIÓN DEL MODELO DE CALIDAD 88
4.2 SELECCIÓN DE CARACTERÍSTICAS, SUBCARACTERÍSTICAS Y MÉTRICAS A EVALUAR DEL SOFTWARE DE MONITOREO 88
4.3 OBTENCIÓN DE RESULTADOS DE LAS MÉTRICAS DE CALIDAD EXTERNA DEL SISTEMA 90
4.3.1 FUNCIONALIDAD *90*
4.3.2 CONFIABILIDAD *92*
4.3.3 USABILIDAD *92*
4.3.4 EFICIENCIA *95*
4.3.5 MANTENIBILIDAD *95*
4.3.6 PORTABILIDAD *98*
4.4 DETERMINACIÓN DE LA CALIDAD DEL SISTEMA 99
4.5 INTERPRETACIÓN DE LOS RESULTADOS 100

CAPITULO V **104**

COSTOS Y BENEFICIOS **104**

5.1. EVALUACION DE COSTOS DE PROGRAMACION 104
5.2 COSTO DE ELABORACION DEL PROYECTO 106
5.3 BENEFICIOS 107

CAPITULO VI **108**

SEGURIDAD **108**

6.1. ACCESO AL SISTEMA Y A SUS RECURSOS 108

6.2. MANTENIMIENTO DE LA INTEGRIDAD Y CONFIDENCIALIDAD DE DATOS 109
6.3. COPIA DE SEGURIDAD Y RECUPERACIÓN DE DATOS 109
6.4. RECUPERACIÓN Y REANUDACIÓN DE TRABAJOS 110

CAPITULO VII 111

CONCLUSIONES Y RECOMENDACIONES 111

7.1 CONCLUSIONES 111
7.2 RECOMENDACIONES 112

ANEXOS 114

ANEXO 1 115
ANEXO 2 116
ANEXO 3 117

INDICE DE FIGURAS

CAPITULO II

FIGURA 2. 1. ORGANIGRAMA DEL G.A.M.C 19
FIGURA 2. 2 (FLUJO DE TRABAJO XP) 23
FIGURA 2. 3 PRACTICAS DE XP 25
FIGURA 2. 4 TARJETA CRC 31
FIGURA 2. 5 MODELO DE DATOS 37
FIGURA 2. 6 MODELO DE HIPERTEXTO 38
FIGURA 2. 7 COMPARACIÓN DEL MODELO DE APLICACIÓN WEB CLÁSICO CON AJAX 40
FIGURA 2. 8 CALIDAD DEL PRODUCTO SOFTWARE Y ESTÁNDAR ISO 9126 43
FIGURA 2. 9 CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DE LA CALIDAD DE UN PRODUCTO SOFTWARE 44

CAPITULO III

FIGURA 3. 1 MODELO DE DATOS 69
FIGURA 3. 2 MODELO FÍSICO DE DATOS 70
FIGURA 3. 3 DIAGRAMA DE HIPERTEXTO: REGISTRO DE USUARIOS 71
FIGURA 3. 4 DIAGRAMA DE HIPERTEXTO: REPORTES DE USUARIOS 71
FIGURA 3. 5 DIAGRAMA DE AUTENTIFICACIÓN DE USUARIO 72
FIGURA 3. 6 DIAGRAMA REGISTRO DE USUARIOS NUEVOS 72
FIGURA 3. 7 DIAGRAMA DE HIPERTEXTO: ADMINISTRACIÓN DE USUARIOS. 73
FIGURA 3. 8 DIAGRAMA LISTA DE USUARIOS REGISTRADOS 74
FIGURA 3. 9 DIAGRAMA FICHA REGISTRO DE USUARIO. 74
FIGURA 3. 10 MODELO DE HIPERTEXTO: REGISTRO DE COMUNIDADES 75
FIGURA 3. 11 MODELO DE HIPERTEXTO: MODIFICAR COMUNIDADES. 75
FIGURA 3. 12 MODELO DE HIPERTEXTO: CREACIÓN DE EVENTOS. 76
FIGURA 3. 13 MODELO DE HIPERTEXTO: DETALLE DE EVENTO 76
FIGURA 3. 14 MODELO DE HIPERTEXTO: ASIGNAR AUSPICIADOR EVENTO 77
FIGURA 3. 15 DIAGRAMA LISTADO DE COMUNIDADES. 77
FIGURA 3. 16 DIAGRAMA FORMULARIO REGISTRO COMUNIDADES. 78
FIGURA 3. 17 DIAGRAMA FORMULARIO REGISTRO DE EVENTO. 78
FIGURA 3. 18 DIAGRAMA LISTADO DE EVENTOS A REALIZARSE 79
FIGURA 3. 19 DIAGRAMA REGISTRO DE AUSPICIADORES DE UN EVENTO. 79
FIGURA 3. 20 MODELO DE HIPERTEXTO: BUSCADOR DE CONTENIDOS. 80
FIGURA 3. 21 MODELO DE HIPERTEXTO: INDICADORES ESTADÍSTICAS (REPORTES). 81
FIGURA 3. 22 PANTALLA DE AUTENTIFICACIÓN DE USUARIO 82
FIGURA 3. 23 PANTALLA PRINCIPAL USUARIO AUTENTIFICADO 82
FIGURA 3. 24 PANTALLA FORMULARIO DE REGISTRO USUARIOS 83
FIGURA 3. 25 FICHA DE REGISTRO DE USUARIO 84
FIGURA 3. 26 PANTALLA LISTADO DE EVENTOS A REALIZARSE 84
FIGURA 3. 27 PANTALLA DE REGISTRO DE EVENTOS 85
FIGURA 3. 28 PANTALLA MÓDULO DE REPORTES Y 4 TIPOS DE EXPORTACIÓN 85

CAPITULO IV

FIGURA 4. 1 TABULACIÓN DE RESULTADOS DE LA MEDICIÓN DE CALIDAD 99
FIGURA 4. 2 RESULTADO DEL NIVEL DE CALIDAD DEL SISTEMA DE SEGUIMIENTO Y MONITOREO 100
FIGURA 4. 3 RESULTADOS DEL NIVEL DE CALIDAD DE FUNCIONALIDAD 100

FIGURA 4. 4 RESULTADOS DEL NIVEL DE CALIDAD DE CONFIABILIDAD 101
FIGURA 4. 5 RESULTADOS DEL NIVEL DE CALIDAD DE USABILIDAD 101
FIGURA 4. 6 RESULTADOS DEL NIVEL DE CALIDAD DE EFICIENCIA 102
FIGURA 4. 7 RESULTADOS DEL NIVEL DE CALIDAD DE MANTENIBILIDAD 102
FIGURA 4. 8 RESULTADOS DEL NIVEL DE CALIDAD DE PORTABILIDAD 103

CAPITULO VI

FIGURA 6. 1 DIAGRAMA VALIDACIÓN DE USUARIO 109

INDICE DE TABLAS

CAPITULO II

TABLA 2. 1 EJEMPLO DE MÉTRICAS EXTERNAS (MÉTRICAS DE ADECUACIÓN) 46
TABLA 2. 2 EJEMPLO DE MÉTRICAS EXTERNAS (MÉTRICAS DE ADECUACIÓN) 48

CAPITULO III

TABLA 3. 1 INTERRELACIÓN DE METODOLOGÍAS 50
TABLA 3. 2 HISTORIA DE USUARIO NRO. 1 51
TABLA 3. 3 HISTORIA DE USUARIO NRO. 2 51
TABLA 3. 4 HISTORIA DE USUARIO NRO. 3 52
TABLA 3. 5 HISTORIA DE USUARIO NRO. 4 53
TABLA 3. 6 HISTORIA DE USUARIO NRO. 5 53
TABLA 3. 7 HISTORIA DE USUARIO NRO. 6 54
TABLA 3. 8 HISTORIA DE USUARIO NRO. 7 54
TABLA 3. 9 HISTORIA DE USUARIO NRO. 8 55
TABLA 3. 10 TARJETA DE TAREA DESARROLLO MODULO DE REGISTRO DE USUARIOS 55
TABLA 3. 11 TARJETA DE TAREA IMPLEMENTACIÓN DE LA BASE DE DATOS 56
TABLA 3. 12 TARJETA DE TAREA: DESARROLLO DE LA INTERFAZ 56
TABLA 3. 13 TARJETA DE TAREA REPORTE DE USUARIOS 57
TABLA 3. 14 TARJETA DE TAREA DESARROLLAR MÓDULO DE ADMINISTRADOR 57
TABLA 3. 15 MODULO CONFIGURACIÓN DEL SISTEMA 58
TABLA 3. 16 IMPLEMENTACIÓN DE TABLAS MODULO ADMINISTRADOR 58
TABLA 3. 17 DESARROLLO DE LA INTERFAZ DE ADMINISTRADOR 59
TABLA 3. 18 UTILIZACIÓN DE JQUERY MODULO ADMINISTRADOR 59
TABLA 3. 19 MODULO REGISTRO DE COMUNIDADES 60
TABLA 3. 20 TARJETA DE TAREA: INTERFAZ DE USUARIO MODULO REGISTRO DE COMUNIDADES 60
TABLA 3. 21 TARJETA DE TAREA: DESARROLLO SUB MODULO REGISTRO DE EVENTOS CULTURALES 61
TABLA 3. 22 TARJETA DE TAREA: DESARROLLO INTERFAZ SUB MODULO REGISTRO DE EVENTOS CULTURALES 61
TABLA 3. 23 TARJETA DE TAREA: DESARROLLO SUB MODULO REGISTRO DE AUSPICIADORES 62
TABLA 3. 24 TARJETA DE TAREA: IMPLEMENTAR HERRAMIENTAS DE BÚSQUEDA 62
TABLA 3. 25 TARJETA DE TAREA: DESARROLLAR MÓDULO DE REPORTES 63
TABLA 3. 26 TARJETA DE TAREA: IMPLEMENTAR HERRAMIENTAS ESTADÍSTICAS 63
TABLA 3. 27 PLAN DE ENTREGAS 64
TABLA 3. 28 PLANIFICACIÓN DE ITERACIONES 65
TABLA 3. 29 METÁFORA DEL SISTEMA 66
TABLA 3. 30 TARJETA C.R.C. USUARIOS. 66
TABLA 3. 31 TARJETA C.R.C. AUSPICIADORES 67
TABLA 3. 32 TARJETA C.R.C. COMUNIDADES. 67
TABLA 3. 33 TARJETA C.R.C. EVENTOS. 68
TABLA 3. 34 TARJETA C.R.C. REPORTES. 68
TABLA 3. 35 VERIFICACIÓN DE REGISTRO DE USUARIO 87
TABLA 3. 36 VERIFICACIÓN MODULO EVENTOS 87

CAPITULO IV

TABLA 4. 1 CARACTERÍSTICAS Y SUBCARACTERISTÍCAS SELECCIONADAS PARA LA EVALUACIÓN 90

TABLA 4. 2 RESULTADOS DE LAS MÉTRICAS DE ADECUACIÓN 91
TABLA 4. 3 RESULTADOS DE LAS MÉTRICAS DE ADECUACIÓN 92
TABLA 4. 4 RESULTADOS DE LAS MÉTRICAS DE MADUREZ 92
TABLA 4. 5 RESULTADOS DE LAS MÉTRICAS DE ENTENDIMIENTO 93
TABLA 4. 6 RESULTADOS DE LAS MÉTRICAS DE APRENDIZAJE 94
TABLA 4. 7 RESULTADOS DE LAS MÉTRICAS DE OPERACIÓN 95
TABLA 4. 8 RESULTADOS DE LAS MÉTRICAS DE ATRACTIBILIDAD 95
TABLA 4. 9 RESULTADOS DE LAS MÉTRICAS DE COMPORTAMIENTO EN EL TIEMPO 95
TABLA 4. 10 RESULTADOS DE LAS MÉTRICAS DE ANALIZABILIDAD 96
TABLA 4. 11 RESULTADOS DE LAS MÉTRICAS DE ESTABILIDAD 97
TABLA 4. 12 RESULTADOS DE LAS MÉTRICAS DE FACILIDAD DE PRUEBA 98
TABLA 4. 13 RESULTADOS DE LAS MÉTRICAS DE ADAPTABILIDAD 98

CAPITULO V

TABLA 5. 1 VALORES DE LOS COEFICIENTES A Y B DEL MODELO COCOMO POR TIPO DE PROYECTO 104
TABLA 5. 2 VALORES DE LOS COEFICIENTES C Y D DEL MODELO COCOMO POR TIPO DE PROYECTO 105
TABLA 5. 3 TABLA DE COSTOS DEL SISTEMA 107

CAPITULO VI

TABLA 6. 1 CONSECUENCIAS DE INTERRUPCIONES EN EL SOFTWARE 110

CAPITULO I

MARCO REFERENCIAL

1.1 INTRODUCCION

La información es muy importante para las organizaciones públicas y privadas, las cuales han reconocido la importancia de la misma colocándola en un primer lugar como un recurso principal para la toma de decisiones. La administración correcta de este trae beneficios como: la entrega de información oportuna, apoyo en la toma de decisiones, menos recursos económicos en personal y en material de escritorio.

Este suceso se materializa en la implementación de sistemas informáticos dentro de las organizaciones públicas como ser los Gobiernos Municipales, proveyendo de esta manera importancia al control, seguimiento y análisis de información que para muchas de estas organizaciones constituye una utilidad más para ellas, en la gestión, manipulación, almacenamiento de la información.

Por estas razones, el hecho de implementar un software que se adecue a los requerimientos y necesidades de la organización, se convierte no solo en una necesidad sino también en una obligación de quienes están a cargo de facilitar la labor de su personal.

Debido a estas razones, la Dirección General de Cultura del Gobierno Autónomo Municipal de Copacabana, pone en marcha el desarrollo de un Software orientado a la web apto para el manejo de eventos culturales de todas las comunidades pertenecientes a este municipio de esta manera el personal administrativo obtendrá información necesaria oportunamente.

El presente proyecto de grado trata del desarrollo e implementación de un Software de seguimiento y monitoreo de eventos culturales para la dirección General de cultura del Gobierno Autónomo Municipal de Copacabana, aplicando elementos de metodologías de desarrollo de software y tecnologías web.

1.2 PROBLEMA

1.2.1 ANTECEDENTES

a) ANTECEDENTES DEL PROYECTO

Algunos proyectos de grado realizados por estudiantes de la carrera de Informática, llevan o proveen aportes interesantes y tratan temas relacionados al presente proyecto.

Rudy Idiáquez Rocha (2008), Sistema Automatizado para el Seguimiento, Evaluación y Toma de Decisiones en Proyectos. Fue desarrollado para el Instituto de Genética de la Facultad de Medicina de la Universidad Mayor de San Andrés; está orientada a facilitar el monitoreo de los proyectos que se desarrollan dentro del Instituto, permitiendo realizar un seguimiento del avance y estado de los mismos. Su enfoque es muy general, pudiéndose aplicar a varios tipos de proyectos.

Lady Gonzales Quispe (2009), Sistema de Monitoreo y Control para el Programa Tomando Decisiones – Save The Children. Este sistema fue desarrollado con el fin de contribuir con los objetivos planteados en el programa, teniendo como propósito descubrir fortalezas y/o debilidades para establecer líneas de acción y tomar decisiones de carácter estratégico.

b) LA INSTITUCION

El Gobierno Autónomo Municipal de Copacabana es una institución pública perteneciente a la provincia Manco Kapac del departamento de La Paz provincia y tiene cerca de 6 mil habitantes el cual abarca 33 comunidades campesinas originarias, con un total aproximado de 15 mil habitantes, y por ser una región turística realiza diversas actividades culturales de distinta índole ya sea para promover las tradiciones de la región o generar ingresos económicos..

La Dirección General de Cultura dependiente del Gobierno Autónomo de Copacabana es la encargada de organizar, administrar, fomentar diferentes actividades culturales de todas las comunidades pertenecientes a este municipio,. ya que este municipio es una de las principales atracciones turísticas de Bolivia y son la fuente principal de ingreso de la población de esta región.

La Dirección General de Cultura actualmente no cuenta con un Software para organizar ni monitorear o ver estadísticas de eventos realizados o por realizarse ya que toda la

información se la hace de forma manual y esto provoca en la mayoría de los casos una mala toma de decisiones o mala organización de los mismos.

1.2.2 PLANTEAMIENTO DEL PROBLEMA

El sistema actual para el monitoreo de eventos culturales es manual ya que el equipo responsable de monitoreo, recolecta información mediante instrumentos físicos como solicitudes, formularios e informes de los encargados. Esta forma de realizar el monitoreo, conlleva problemas como:

- El tiempo empleado para recolectar la información es muy largo.
- Dificultad para procesar oportunamente los datos.
- Demora en el análisis de datos para la retroalimentación.
- Demora en la entrega de informes a los ejecutivos.
- Extravió de documentación.
- La organización de la información es inadecuada.
- Dificultad para realizar estadísticas en base a eventos pasados.
- La búsqueda de información requerida es morosa.

1.2.3 FORMULACIÓN DEL PROBLEMA

Por lo expuesto anteriormente se plantea el siguiente problema:

La ineficiente administración de la información de la Dirección General de Culturas del Gobierno Autónomo Municipal de Copacabana no permite el manejo efectivo de dicha información llegando a ser burocrático e ineficiente el cual no permite una adecuada toma de decisiones.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Desarrollar e implementar un Software orientado a la web para el seguimiento y monitoreo de los eventos culturales que realiza la Dirección General de Cultura del Gobierno Autónomo Municipal de Copacabana para poder agilizar el proceso de recolección de información necesaria para su análisis y retroalimentación oportuna.

1.3.2 OBJETIVOS ESPECÍFICOS.

- Desarrollar un Sistema de Información vía Web aplicando elementos de metodologías de desarrollo de software.
- Clasificar la información necesaria para el monitoreo de todos los eventos organizados por la Dirección General Cultura.
- Facilitar el acceso de la información a los usuarios del Sistema, de acuerdo al un rol asignado, de manera que sea rápido y preciso.
- Desarrollar instrumentos computarizados de recolección de información, tales como cuestionarios y formularios electrónicos, para el monitoreo de las actividades de la Dirección General de Cultura.
- Generar reportes e informes dirigidos a los encargados de monitoreo y encargados de los eventos.

1.4 JUSTIFICACION

El desarrollo de este Software facilitara al personal de equipo técnico responsable de monitoreo de la Dirección General de Cultura del Gobierno Autónomo Municipal de Copacabana, pues la esencia del mismo es facilitar el acceso a la información para el monitoreo y realizar un análisis oportuno para la retroalimentación y toma de decisiones, también optimizar el uso de recursos y material de escritorio.

Económicamente el proyecto es viable para su elaboración e implementación ya que el Gobierno Autónomo Municipal de Copacabana cuenta con los equipos computacionales necesarios lo cual no conlleva gastos adicionales, además ya que los procesos estarán automatizados el cual permite reducir el tiempo de acceso a la información, reducir el excesivo manejo de documentación y además reducir el costo excesivo en el uso de los recursos económicos. El desarrollo e implementación del presente proyecto, permitirá la optimización del material utilizado por el equipo responsable de organización y monitoreo de eventos culturales de la Dirección General de Cultura del G.A.M.C.

1.5 LIMITES Y ALCANCES

El Software de Seguimiento y Monitoreo de Eventos Culturales a nivel institucional tiene un alcance en la Dirección General de Cultura del Gobierno Autónomo Municipal de Copacabana, para facilitar el proceso de recolección de información y la generación de informes para un análisis y retroalimentación oportuna, así como también llevar un registro de los resultados obtenidos en todas las gestiones.

A nivel geográfico el proyecto tiene un alcance de las 33 comunidades pertenecientes a este municipio y pretende proveer información de todo el proceso de organización y ejecución de eventos culturales para contar con datos fidedignos y oportunos, así como también poder administrar sus propias actividades mediante el personal representante de la comunidad en la Dirección General de Cultura y poder llevar un registro de todas las gestiones.

1.6 APORTES

El aporte de la implementación del presente proyecto es brindar un Software que mejorara el proceso de recolección de información para el monitoreo y la capacidad de procesamiento de mayor cantidad de datos para el análisis y retroalimentación oportuna de la institución.

El desarrollo e implementación del presente proyecto de grado es importante, porque beneficiara a la Dirección General de Cultura del G.A.M.C. Mejorando en gran medida el tiempo de los procesos efectuados en esta tarea, mediante la automatización de los procesos manuales, que se llevan a cabo dentro la misma.

CAPITULO II

MARCO TEORICO

2.1 MARCO INSTITUCIONAL

2.1.1 MISION

Dirección General de Cultura del Gobierno Autónomo Municipal de Copacabana es una institución de servicio proactiva, moderna, eficiente y eficaz, administrada de manera transparente, que contribuye a mejorar las condiciones de vida de la población que abarca 33 comunidades tiene como función realizar y promover eventos culturales para difundir la cultura de la región, también para la obtención de recursos, fomentando el turismo en base a la ejecución de programas y actividades definidas a través de un proceso de planificación participativa y estratégica. [Memorias G.A.M.C.]

2.1.2 VISION

La Provincia Manco Kapac de la ciudad de La Paz, se constituye en el centro turístico, cultural y productivo de la región, y del país, integrado nacional e internacionalmente, con una elevada calidad de vida que incrementa día a día la felicidad y prosperidad de sus habitantes.

Los atractivos turísticos de la región y los aspectos culturales de sus habitantes, permiten a los visitantes compartir y generar empatías lo que hace de Copacabana un centro de recepción de un número cada vez mayor de visitantes.

Elevada calidad de vida, donde gracias a la actividad cultural, turística y trabajo de sus habitantes, se logra tener un nivel de ingresos que permite tener acceso a los bienes y servicios para el bienestar de las personas. [Memorias G.A.M.C.]

2.1.3 ORGANIGRAMA

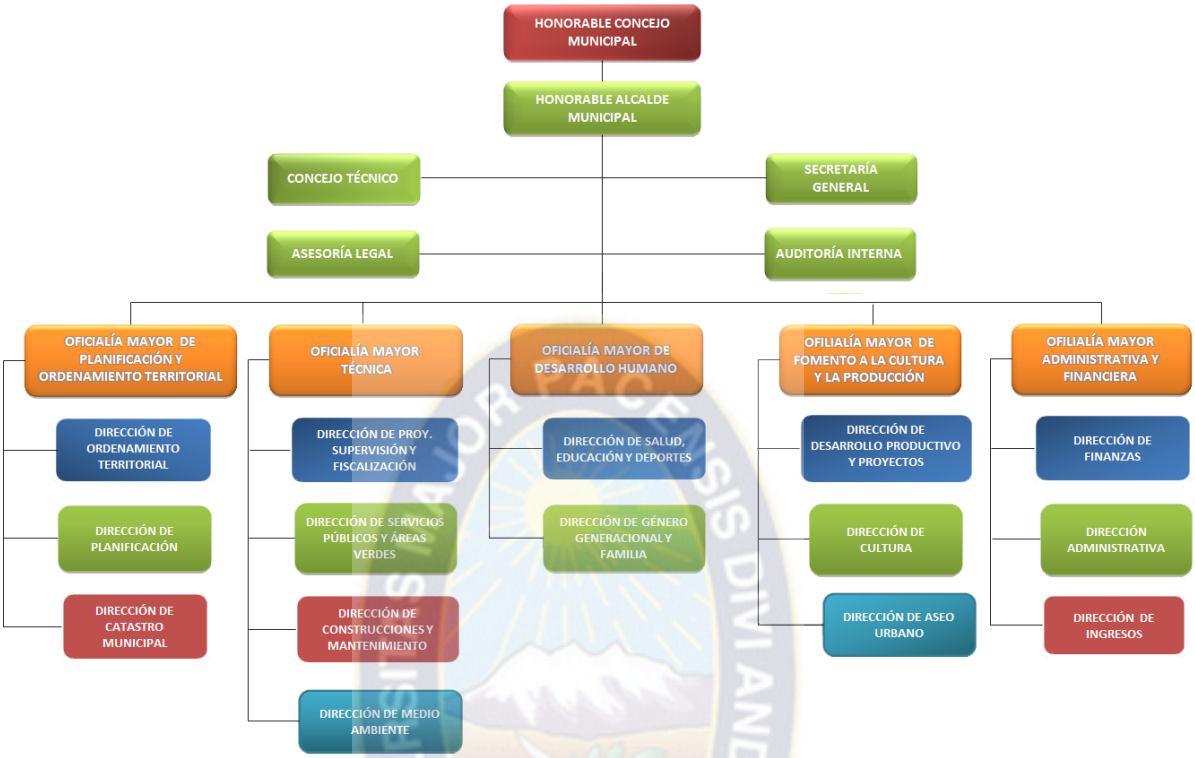


Figura 2. 1. Organigrama del G.A.M.C.

Fuente: Memoria Institucional – G.A.M.C.

En la figura 2.1, se puede apreciar la estructura organizacional del Gobierno Autónomo Municipal de Copacabana (G.A.M.C.), a continuación se describen las áreas más importantes dentro de esta estructura.

Honorable Concejo Municipal la máxima instancia de decisión sus principales funciones según el artículo 313 de la Constitución son:

- Reglamentar las funciones y la eficiente prestación de los Servicios a cargo del municipio.
- Adoptar los correspondientes planes y programas de Desarrollo económico y social y de obras públicas.

- Autorizar al alcalde para celebrar contratos y ejercer pro Tempore precisas funciones de las que corresponden al concejo.
- Dictar las normas orgánicas del presupuesto y expedir anualmente el presupuesto de rentas y gastos.
- Dictar las normas necesarias para el control, la preservación y Defensa del patrimonio ecológico y cultural del municipio.
- Las demás que la Constitución y la ley les asignen.

2.1.4 ASPECTOS HISTORICOS Y CULTURALES

La región donde se encuentra la actual población de Copacabana, considerada como la cuna de la cultura Aymara, estuvo poblada desde siglos atrás. Indicios arqueológicos sugieren que los pobladores conocidos más antiguos de esta zona pertenecieron a la cultura Chiripa, que aparece en el siglo XIV a.C., (alrededor del año 1.300 a.C.). Luego de la cultura Chiripa, se desarrolla la Cultura Tiwanakota, primer gran imperio en estas tierras, que cubría los actuales territorios de Perú, Chile y Bolivia. Sus orígenes se remontan al año 1.580 a.C., y por razones aún en discusión

Copacabana es un centro de peregrinaje, Santuario y sitio de devoción religiosa. También es un centro de descanso de fin de semana y días feriados, para disfrutar de las maravillosas vistas y el entretenimiento que la región ofrece. [Memorias G.A.M.C.]

2.2 DESCRIPCION DE LA DIRECCION DE CULTURA

La Dirección General de Cultura dependiente del Gobierno Autónomo Municipal de Copacabana tiene el objeto de proteger, fomentar y promocionar el dialogo intercultural y desarrollo cultural y creativo, así como defender el patrimonio cultural del Municipio, diseñando políticas de fomento, promoción, formación y difusión local, nacional e internacional de valores culturales, apoyando y facilitando la provisión de infraestructura y medios que permitan su expansión. [Memorias G.A.M.C.]

2.2.1 FUNCIONES Y ATRIBUCIONES ESPECÍFICAS

- Desarrollar políticas, normas y programas de regulación, protección, recuperación, catalogación, conservación y promoción del patrimonio cultural y natural.

- Diseñar y ejecutar programas para la defensa y valorización del patrimonio cultural y natural.
- Realizar acciones de defensa ante cualquier tipo de apropiación indebida.
- Organizar, supervisar y coordinar el desarrollo de actividades culturales en fechas cívicas e históricas Paceñas y Bolivianas.
- Diseñar y ejecutar campañas y servicios de información y promoción de valores culturales e históricos.
- Realizar, promover y coordinar el desarrollo de investigaciones que promuevan el análisis y conocimiento de valores culturales.
- Fomentar la construcción de espacios de encuentro, infraestructura y equipamiento cultural para el desarrollo de actividades artísticas culturales.
- Administrar y controlar el funcionamiento de los espacios culturales municipales [museos, bibliotecas, espacios escénicos, casas distritales de las culturas, auditorios y salas de exposición.

2.3 MONITOREO

Monitoreo es un proceso continuo de recojo y análisis de información con la finalidad de medir el progreso del proyecto/programa hacia el logro de los resultados (usos, efectos e impactos) esperados y mejorar la toma de decisiones referidos a las siguientes etapas.

2.3.1 EFECTOS E IMPACTOS

EFECTO: Cambios a corto plazo (efecto directo), que se relacionan directamente con el buen uso que el grupo meta hace de los productos que el proyecto genera. Quiere decir: el efecto resulta del buen uso de los productos.

IMPACTO: Cambios en el largo plazo – intencionales y no intencionales, positivos y negativos – a nivel de los diferentes actores de un proyecto (grupos meta, organizaciones contrapartes, organizaciones locales, etc.) y en el ámbito del proyecto como tal. Estos impactos pueden producirse como consecuencia de las intervenciones realizadas durante la implementación del proyecto y/o una vez finalizado el mismo.

2.3.2 MONITOREO ORIENTADO A EFECTOS E IMPACTOS

Se aplica sistemática y regularmente en el transcurso de un proyecto para observar y documentar los cambios de corto, mediano y largo plazo. El monitoreo orientado a efectos e impactos es un instrumento para la gestión de un proyecto, y a la vez un campo de aprendizaje para todos los involucrados. Sirve para comprobar a buena hora si el proyecto es capaz o no de alcanzar los objetivos definidos. A través de los indicadores de 30 efecto/impacto se observan aquellas partes en una cadena de resultados que responden a una expectativa claramente definida con anticipación.

2.3.3 INDICADORES

Son la base para el sistema de monitoreo orientado a efectos e impactos. Los indicadores deben describir el uso de los productos y las mejoras o cambios que se dan a partir del uso. Los indicadores del nivel de uso son mucho más fáciles de formular y revisar que indicadores a nivel de efectos e impactos. Los indicadores deben precisar: actores, cantidad, calidad, lugar y fecha.

Criterios de calidad para indicadores

Un indicador tiene que ser:

Claro: Se entiende fácilmente sin necesidad de explicación adicional. Relevante: Mide un aspecto central de un cambio deseado (relación estrecha con el objetivo del proyecto).

Preciso: Incorpora dimensiones cualitativas, cuantitativas, de tiempo y de lugar, y determina el grupo meta.

Realista: ¿Los valores mencionados en el indicador realmente son alcanzables en el periodo de tiempo previsto?

Verificable: ¿Tiene un costo aceptable levantar los datos necesarios para medir el indicador?

2.3.4 EVALUACIÓN

Apreciación sistemática y objetiva de un proyecto en curso o concluido, de su diseño, su puesta en práctica y sus efectos e impactos. El objetivo es determinar la pertinencia y el logro de los resultados, así como la eficiencia, la eficacia, el impacto y la sostenibilidad.

2.4 METODOLOGIA DE DESARROLLO DEL SISTEMA

En esta parte del documento se hace una presentación teórica de XP como metodología de desarrollo y de los principios teóricos que inspiraron esta metodología. Finalmente se entra en detalle con cada una de las etapas de desarrollo (planeación, diseño, codificación y pruebas) describiendo cada uno de los aspectos que la componen.

2.4.1 METODOLOGIA DE DESARROLLO XP (eXtreme Programming)

XP resalta una serie de valores y principios que deben tenerse en cuenta y practicarlos durante el tiempo de desarrollo que dure el proyecto. Al final de este apartado se enuncian algunas de las características que deben tener los proyectos que se realicen con XP.

La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance. Además, se especifica que, de estas cuatro variables, sólo tres de ellas podrán ser fijadas arbitrariamente por actores externos al grupo de desarrolladores (clientes y jefes de proyecto). El valor de la variable restante podrá ser establecido por el equipo de desarrollo, en función de los valores de las otras tres.

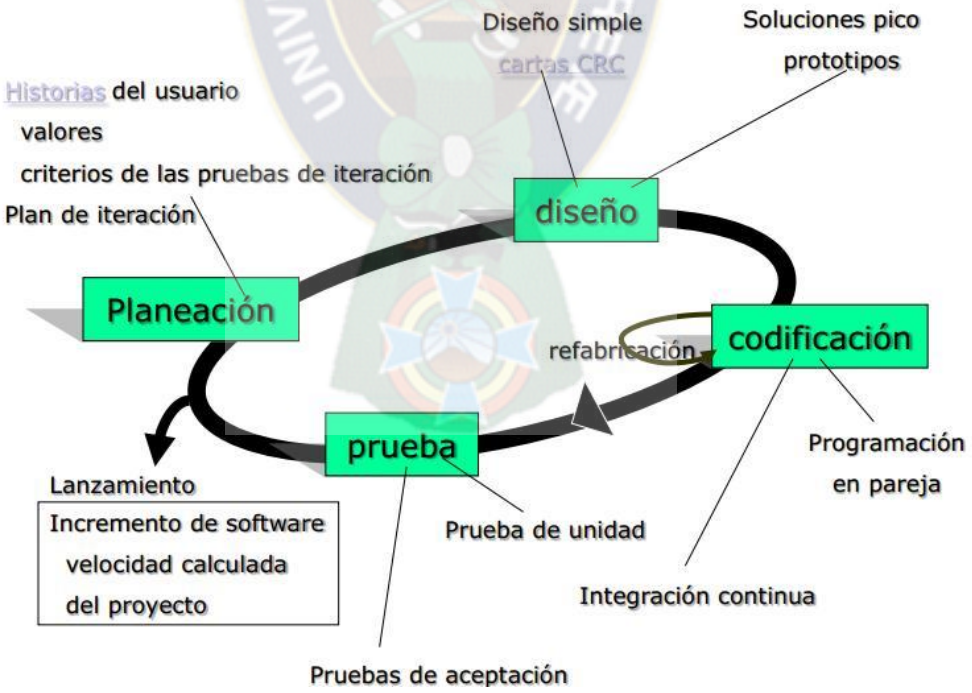


Figura 2. 2 (Flujo de trabajo XP)

Fuente: [ONESS, 2005]

a) VALORES DE XP

Más que una metodología, XP se considera una disciplina, la cual está sostenida por valores y principios propios de las metodologías ágiles. Existen cuatro valores que cumplen su papel como pilares en el desarrollo de las metodologías livianas:

La comunicación. En la metodología XP es muy importante que exista un ambiente de colaboración y comunicación al interior del equipo de desarrollo, así como en la interacción de éste con el cliente. En XP la interacción con el cliente es tan estrecha, que es considerado parte del equipo de desarrollo.

La simplicidad. Este valor se aplica en todos los aspectos de la programación extrema. Desde diseños muy sencillos donde lo más relevante es la funcionalidad necesaria que requiere el cliente, hasta la simplificación del código mediante la refactorización del mismo. La programación XP no utiliza sus recursos para la realización de actividades complejas, sólo se desarrolla lo que el cliente demanda, de la forma más sencilla.

La retroalimentación. Se presenta desde el comienzo del proyecto, ayuda a encaminarlo y darle forma. Ésta se presenta en los dos sentidos, por parte del equipo de trabajo hacia el cliente, con el fin de brindarle información sobre la evolución del sistema, y desde el cliente hacia el equipo en los aportes a la construcción del proyecto.

El coraje. El equipo de desarrollo debe estar preparado para nfrentarse a los continuos cambios que se presentarán en el transcurso de la actividad. Cada integrante de be tener el valor de exponer los problemas o dudas que halle en la realización del proyecto. Aún con estas variaciones, las jornadas de trabajo deben proporcionar el máximo rendimiento.

b) PRACTICAS DE XP

A partir de los valores se plantea una serie de prácticas que sirven de guía para los desarrolladores en esta metodología. Una de los aspectos más importantes para XP son las doce reglas que se plantean, las cuales se caracterizan por su grado de simplicidad y por su enfoque en la practicidad, además de que cada regla se complementa con las demás. A continuación se realizará una breve descripción de cada una de ellas.

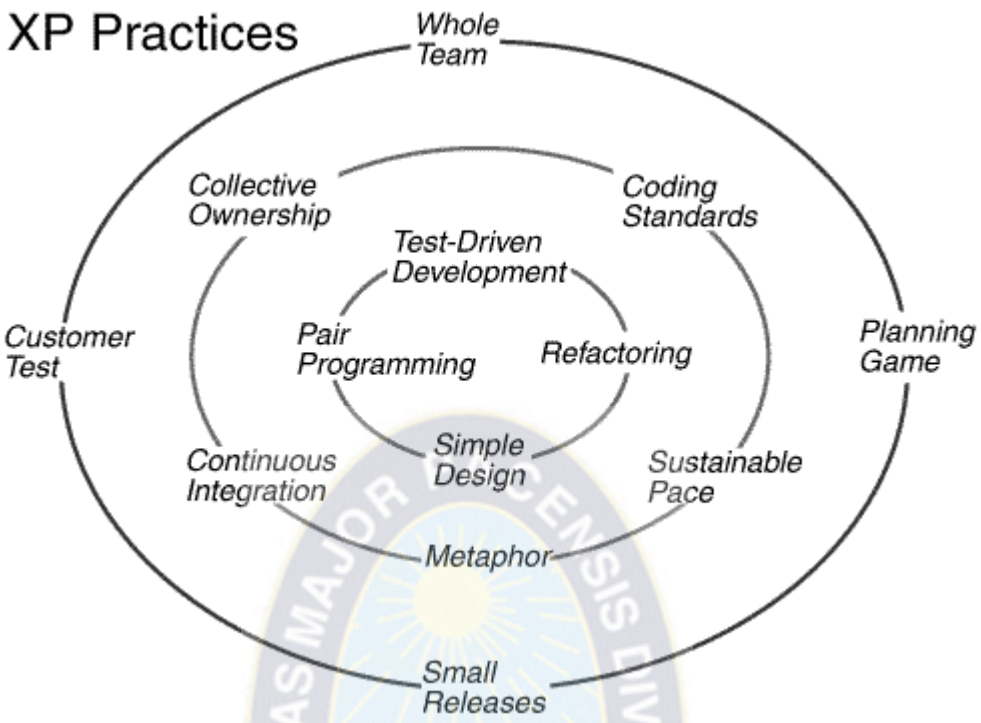


Figura 2. 3 Practicas de XP

Fuente [ONESS, 2005]

El desarrollo está dirigido por pruebas. Antes de realizar una unidad de código, es necesario contar con su respectiva unidad de pruebas. El programador realiza pruebas dirigidas al funcionamiento de nuevas adiciones o módulos al sistema. El cliente con ayuda del tester se encarga de diseñar las pruebas de aceptación, cuyo propósito es verificar que las historias de usuario se hayan implementado correctamente.

El juego de la planificación. Desde el comienzo del desarrollo se requiere que el grupo y el cliente tengan una visión general y clara del proyecto, es decir, deben entender y estar de acuerdo con lo que el "otro" plantee. En el transcurso del proyecto se realizan diferentes reuniones, con el fin de organizar las tareas e ideas que surgen tanto por parte del cliente como por el equipo.

Cliente in-situ. El cliente, o un representante del mismo, deben estar en el sitio de desarrollo para solucionar las preguntas o dudas que se puedan presentar a medida que se realice el proyecto.

Programación en parejas. XP propone que exista una pareja de programadores por monitor y teclado, como medida para aumentar la calidad del código. Esta práctica busca reducir los errores de codificación, mientras uno de los programadores busca una forma de dar funcionalidad a un módulo, el otro programador aprueba dicho código y busca la forma de simplificarlo.

Entregas pequeñas. En la programación extrema se realizan entregas constantes de módulos funcionales completos, de tal forma que en todo momento el cliente tiene una parte de aplicación funcionando. En XP no existe el desarrollo incompleto de una tarea, ésta se ejecuta en su totalidad o no se hace.

Refactorización sin piedad. El código se revisa de forma permanente para depurarlo y simplificarlo, buscando la forma de mejorarlo. La refactorización se realiza durante todo el proceso de desarrollo.

Integración continúa del código. El código de los módulos debe ser integrado a cortos plazos de tiempo, preferiblemente no mayores a un día. Esto facilita la búsqueda y la corrección de errores de codificación e integración que se presenten en el proceso.

Diseño simple. Sólo se realiza lo necesario para que la aplicación cumpla con la funcionalidad requerida por el cliente. No es conveniente realizar diseños complejos que posiblemente no aporten soluciones claras al proyecto, y que a la hora de cambiar los requerimientos se conviertan en una gran barrera de tiempo.

Utilización de metáforas del sistema. Para el mejor entendimiento de los elementos del sistema por parte del equipo de desarrollo se acude a la utilización de metáforas, como una forma de universalizar el lenguaje del sistema.

Propiedad colectiva del código. El código no es conocido por una sola persona del grupo de trabajo, esto facilita implementar cambios al programa por parte de otros integrantes del equipo.

Convenciones de código. La aplicación de estándares de programación al código fuente de la aplicación, permite que todas las personas que conforman el grupo de trabajo puedan entender y realizar modificaciones al código del sistema.

No trabajar horas extras. Es preferible volver a estimar los tiempos de entrega. Con esta práctica se busca utilizar al máximo el rendimiento y energía del programador.

c) ALCANCE DE XP

La programación extrema es conveniente en ciertas situaciones, pero también es necesario saber que presenta controversia en otras. Esta metodología es aplicable con resultados positivos a proyectos de mediana y pequeña envergadura, donde los grupos de trabajo no superan 20 personas. Los proyectos realizados bajo esta metodología cumplen con lo estrictamente necesario en su funcionalidad en el momento necesario: hacer lo que se necesita cuando se necesita. En XP no es conveniente precipitarse o adelantarse a las tareas que se han establecido previamente sin el consentimiento del cliente, estos hechos conllevan a inyectar complejidad al sistema, alejándolo del concepto de simplicidad.

2.4.2 PLANEACION

La planeación es la etapa inicial de todo proyecto en XP. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema. En este punto se identifican el número y tamaño de las iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto.

En este apartado se tendrán en cuenta ocho elementos, los cuales son los siguientes. Historias de usuario, velocidad del proyecto, iteraciones, entregas pequeñas, reuniones, roles en XP, traslado del personal y ajuste a XP.

a) HISTORIAS DE USUARIO

El sistema es desarrollado para el cliente, por lo tanto, el usuario es quien decide que tareas realizará la aplicación. Este planteamiento se desarrolla a lo largo del proyecto: el cliente es quien decide que hacer. Como primer paso, se debe proporcionar una idea clara de lo que será el proyecto en sí.

Las historias de usuario son utilizadas como herramienta para dar a conocer los requerimientos del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe una actividad que realizará el sistema; la redacción de los mismos se

realiza bajo la terminología del cliente, no del desarrollador, de forma que sea clara y sencilla, sin profundizar en detalles.

Se puede considerar que las historias de usuario en XP juegan un papel similar a los casos de uso en otras metodologías, pero en realidad son muy diferentes. Las historias de usuario sólo muestran la silueta de una tarea a realizarse. Por esta razón es fundamental que el usuario o un representante del mismo se encuentren disponibles en todo momento para solucionar dudas, estas no proporcionan información detallada acerca de una actividad específica.

Las historias de usuario también son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar las entregas. En una entrega se puede desarrollar una o varias historias de usuario, esto depende del tiempo que demore la implementación de cada una de las mismas.

b) VELOCIDAD DEL PROYECTO

Es una medida de la capacidad que tiene el equipo de desarrollo para evacuar las historias de usuario en una determinada iteración. Esta medida se calcula totalizando el número de historias de usuario realizadas en una iteración. Para la iteración siguiente se podrá (teóricamente) implementar el mismo número de historias de usuario que en la iteración anterior

Cabe recordar que la velocidad del proyecto ayuda a determinar la cantidad de historias que se pueden implementar en las siguientes iteraciones, aunque no de manera exacta. La revisión continua de esta métrica en el transcurso del proyecto se hace necesaria, ya que las historias varían según su grado de dificultad, haciendo inestable la velocidad de la realización del sistema.

c) ITERACIONES

En la metodología XP, la creación del sistema se divide en etapas para facilitar su realización. Por lo general, los proyectos constan de más de tres etapas, las cuales toman el nombre de iteraciones, de allí se obtiene el concepto de metodología iterativa. La duración ideal de una iteración es de una a tres semanas.

Para cada iteración se define un módulo o conjunto de historias que se van a implementar. Al final de la iteración se obtiene como resultado la entrega del módulo correspondiente, el cual debe haber superado las pruebas de aceptación que establece el cliente para la verificar el cumplimiento de los requisitos. Las tareas que no se realicen en una iteración son tomadas en cuenta para la próxima iteración, donde se define, junto al cliente, si se deben realizar o si deben ser removidas de la planeación del sistema.

d) ENTREGAS PEQUEÑAS

La duración de una iteración varía entre una y tres semanas, al final de la cual habrá una entrega de los avances del producto, los cuales deberán ser completamente funcionales. Estas entregas deben caracterizarse por ser frecuentes.

e) REUNIONES

El planeamiento es esencial para cualquier tipo de metodología, es por ello que XP requiere de una revisión continua del plan de trabajo. A pesar de ser una metodología que evita la documentación exagerada, es muy estricta en la organización del trabajo.

Plan de entregas. Al comenzar el proyecto se realiza una reunión entre el equipo de trabajo y los clientes. En dicha reunión se define el marco temporal de la realización del sistema. El cliente expone las historias de usuario a los integrantes de grupo, quienes estimarán el grado de dificultad de la implementación de cada historia.

Las historias de usuario son asignadas a las diferentes iteraciones según su orden de relevancia para el proyecto. En el proceso de selección de las historias de usuario para cada iteración, se tiene en cuenta que la suma de las estimaciones sea aproximada a la velocidad del proyecto de la iteración pasada.

Finalmente a partir de las historias de usuario, el cliente plantea las pruebas de aceptación con las cuales se comprueba que cada una de éstas ha sido correctamente implementada.

Inicial de Iteración. Al comenzar una iteración se realiza una reunión de la misma, donde se organizan las actividades de programación a realizar. Las historias de usuario son traducidas a tareas y asignadas a los desarrolladores. Los desarrolladores estiman los tiempos para la realización de las tareas. Estas estimaciones son más exactas que las realizadas en la planeación de entregas, por lo tanto no deben exceder la velocidad de

proyecto de la iteración anterior. De ser así, se consulta con el cliente para determinar que historias de usuario se pospondrán para iteraciones futuras.

Diarias o "stand-up meeting". Estas reuniones se realizan al comenzar la jornada laboral. Todo el equipo de desarrollo se reúne para exponer los problemas e ideas que se estén presentando, esto con el fin que el equipo en conjunto construya una mejor solución.

f) ROLES XP

En esta metodología se utiliza el concepto de roles para organizar quienes se encargaran de cada una de las actividades que deben realizarse en el transcurso del proyecto. Dado que este proyecto es individual será únicamente el jefe de proyecto, que tiene como responsabilidad la dirección y organización de las reuniones que se realizan durante el proyecto.

2.4.3 DISEÑO

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo

Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto. Los aspectos que se tratarán a continuación son: *simplicidad en el diseño*, *metáfora del sistema*, *tarjetas CRC*, *spike solution*, no solucionar antes de tiempo y *refactoring*.

a) SIMPLICIDAD EN EL DISEÑO

Una de las partes más importantes de la filosofía XP es la simplicidad en todos los aspectos. Se considera que un diseño sencillo se logra más rápido y se implementa en menos tiempo, por lo cual esto es lo que se busca. La idea es que se haga el diseño más sencillo que cumpla con los requerimientos de las historias de usuario.

En XP se prefiere tener una descripción del sistema o parte de él, en lugar de una serie de complejos diagramas que probablemente tomen más tiempo y sean menos instructivos.

b) METAFORA DEL SISTEMA

Se trata de plasmar la arquitectura de sistema en una “historia” con la cual se le dé al grupo de desarrollo una misma visión sobre el proyecto además de brindarles un primer vistazo muy completo a los nuevos integrantes del grupo para hacer su adaptación más rápida.

Es muy importante dentro del desarrollo de la metáfora darle nombres adecuados a todos los elementos del sistema constantemente. Esto será de mucha utilidad en fases posteriores del desarrollo para identificar aspectos importantes del sistema.

c) TARJETAS DE CLASE, RESPONSABILIDAD, COLABORACION (CRC)

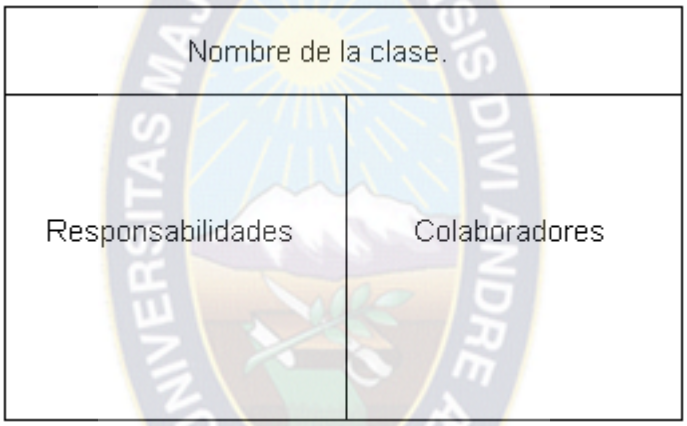


Figura 2. 4 Tarjeta CRC

Fuente: [Ambler, 2002]

La principal funcionalidad que tienen estas, es ayudar a dejar el pensamiento procedimental para incorporarse al enfoque orientado a objetos. Cada tarjeta representa una clase con su nombre en la parte superior, en la sección inferior izquierda están descritas las responsabilidades y a la derecha las clases que le sirven de soporte.

En el proceso de diseñar el sistema por medio de las tarjetas CRC como máximo dos personas se ponen de pie adicionando o modificando las tarjetas, prestando atención a los mensajes que éstas se transmiten mientras los demás miembros del grupo que permanecen sentados, participan en la discusión obteniendo así lo que puede considerarse un diagrama de clases preliminar.

d) REFACTORIZACION (*Refactoring*)

Como se trató al principio de este apartado, el diseño es una tarea permanente durante toda la vida del proyecto y la refactorización concreta este concepto. Como en cualquier metodología tradicional en XP se inicia el proceso de desarrollo con un diseño inicial. La diferencia es que en las metodologías tradicionales este diseño es tan global y completo como se es posible tomando generalmente mucho tiempo en lograrse y con la creencia de que si se ven forzados a modificarlo será un fracaso para el grupo de desarrollo. El caso de XP es el opuesto. Se parte de un diseño muy general y simple que no debe tardar en conseguirse, al cual se le hacen adiciones y correcciones a medida que el proyecto avanza, con el fin de mantenerlo tanto correcto como simple.

2.4.4 CODIFICACIÓN

La codificación es un proceso que se realiza en forma paralela con el diseño y la cual está sujeta a varias observaciones por parte de XP consideradas controversiales por algunos expertos tales como la rotación de los programadores o la programación en parejas.

Además de los mencionados temas, el lector encontrará a continuación una descripción de los siguientes temas: cliente siempre presente, codificar primero la prueba, integración secuencial e integraciones recuentes.

a) CLIENTE SIEMPRE PRESENTE.

Uno de los requerimientos de XP es que el cliente esté siempre disponible. No solamente para solucionar las dudas del grupo de desarrollo, debería ser parte de éste. En este sentido se convierte en gran ayuda al solucionar todas las dudas que puedan surgir, especialmente cara a cara, para garantizar que lo implementado cubre con las necesidades planteadas en las historias de usuario.

b) CODIFICAR PRIMERO LA PRUEBA

Cuando se crea primero una prueba, se ahorra mucho tiempo elaborando el código que la haga pasar, siendo menor el tiempo de hacer ambos procesos que crear el código solamente. Una de las ventajas de crear una prueba antes que el código es que permite identificar los requerimientos de dicho código. De esta forma el desarrollador sabrá con completa certeza en qué momento ha terminado, ya que habrán pasado todas las pruebas.

c) PROGRAMACION EN PAREJAS

Todo el código debe ser creado por parejas de programadores sentados ambos frente a un único computador lo que en principio representa una reducción de un 50% en productividad, sin embargo, según XP no es tal la pérdida. Se entiende que no hay mucha diferencia, en lo que a la cantidad se refiere, entre el código producido por una pareja bajo estas condiciones que el creado por los mismos miembros trabajando en forma separada, con la excepción que uno o ambos programadores sean muy expertos en la herramienta en cuestión.

En este proyecto no se aplicara la programación en parejas por tratarse de un proyecto individual pero será supervisado por un tutor, el cual verificara la correcta aplicación de la metodología.

d) INTEGRACION SECUENCIAL

Uno de los mayores inconvenientes presentados en proyectos de software tiene que ver con la integración, sobre todo si todos los programadores son dueños de todo el código. Para saldar este problema han surgido muchos mecanismos, como darle propiedad de determinadas clases a algunos desarrolladores, los cuales son los responsables de mantenerlas actualizadas y consistentes. Sin embargo, sumado al hecho que esto va en contra de la propiedad colectiva del código no se solucionan los problemas presentados por la comunicación entre clases. XP propone que se emplee un esquema de turnos con el cual solo una pareja de programadores integre a vez. De esta forma se tiene plena seguridad de cuál es la última versión liberada y se le podrán hacer todas las pruebas para garantizar que funcione correctamente. A esto se le conoce como integración secuencial.

e) INTEGRACIONES FRECUENTES

Se deben hacer integraciones cada pocas horas y siempre que sea posible no debe transcurrir más un día entre una integración y otra. De esta forma se garantiza surjan problemas como que un programador trabaje sobre versiones obsoletas de alguna clase. Es evidente que entre más se tarde en encontrar un problema más costoso será resolverlo y con la integración frecuente se garantiza que dichos problemas se encuentre más rápido o aún mejor, sean evitados por completo.

2.4.5 PRUEBAS

XP enfatiza mucho los aspectos relacionados con las pruebas, clasificándolas en diferentes tipos y funcionalidades específicas, indicando quién, cuándo y cómo deben ser implementadas y ejecutadas.

Del buen uso de las pruebas depende el éxito de otras prácticas, tales como la propiedad colectiva del código y la refactorización. Cuando se tienen bien implementadas las pruebas no habrá temor de modificar el código del otro programador en el sentido que si se daña alguna sección, las pruebas mostrarán el error y permitirán encontrarlo. El mismo criterio se aplica a la refactorización. Uno de los elementos que podría obstaculizar que un programador cambie una sección de código funcional es precisamente hacer que esta deje de funcionar. Si se tiene un grupo de pruebas que garantice su buen funcionamiento, este temor se mitiga en gran medida.

a) PRUEBAS UNITARIAS

Estas pruebas se aplican a todos los métodos no triviales de todas las clases del proyecto con la condición que no se liberará ninguna clase que no tenga asociada su correspondiente paquete de pruebas. Uno de los elementos más importantes en estas es que idealmente deben ser construidas antes que los métodos mismos, permitiéndole al programador tener máxima claridad sobre lo que va a programar antes de hacerlo, así como conocer cada uno de los casos de prueba que deberá pasar, lo que optimizará su trabajo y su código será de mejor calidad.

Deben ser construidas por los programadores con el empleo de algún mecanismo que permita automatizarlas de modo tal que tanto su implementación y ejecución consuman el menor tiempo posible permitiendo sacarles el mejor provecho EL empleo de pruebas unitarias completas facilitan la liberación continua de versiones por cuanto al implementar algo nuevo y actualizar la última versión, solo es cuestión de ejecutar de forma automática las pruebas unitarias ya creadas para saber que la nueva versión no contiene errores.

b) PRUEBAS DE ACEPTACION

Las pruebas de aceptación, también llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario. En todas las iteraciones, cada una de las historias de usuario seleccionadas por el cliente deberá tener una o más pruebas de aceptación, de las cuales deberán determinar los casos de prueba e

identificar los errores que serán corregidos. Las pruebas de aceptación son pruebas de caja negra, que representan un resultado esperado de determinada transacción con el sistema. Para que una historia de usuario se considere aprobada, deberá pasar todas las pruebas de aceptación elaboradas para dicha historia. Es importante resaltar la diferencia entre las pruebas de aceptación y las unitarias en lo que al papel del usuario se refiere. Mientras que en las pruebas de aceptación juega un papel muy importante seleccionando los casos de prueba para cada historia de usuario e identificando los resultados esperados, en las segundas no tiene ninguna intervención por ser de competencia del equipo de programadores.

c) CUANDO SE ENCUENTRA UN ERROR

Al momento de encontrar un error debe escribirse una prueba antes de intentar corregirlo. De esta forma tanto el cliente logrará tener completamente claro cuál fue y dónde se encontraba el mismo como el equipo de desarrollo podrá enfocar mejor sus esfuerzos para solucionarlo. Por otro lado se logrará evitar volver a cometerlo. Si el error fue reportado por el cliente y este creó la correspondiente prueba de aceptación junto al equipo de desarrollo, el programador encargado podrá a su vez producir nuevas pruebas unitarias que le permita ubicar la sección específica donde el error se encuentra. (**Fuente:** Reglas y Prácticas en eXtreme Programming Ing. José Joskowicz)

2.5 METODOLOGIA DE MODELADO WEB

Las aplicaciones Web tienen particularidades, lo que hace que se puedan plantear modelos específicos o la forma de realizar el proceso de modelado para ser mas precisos y tener mas ventajas. En la actualidad se propusieron muchos tipos de modelos entre los cuales podemos indicar: UWE, WebML, OODHM, etc.

2.6 WEBML (WEB MODELING LANGUAJE)

La metodoloia de modelado WebML apunta a tener un acercamiento en la estructura del diseño intensivo de datos que se manejan en un sitio Web. Un conjunto de modelos integrados ayudan a los diseñadores en la producción del sitio de alta calidad ya que todas las facetas del diseño deben ser y son dirigidas.

Los principales objetivos del proceso de diseño WebML son:

- Expresar la estructura de una aplicación Web con un alto nivel de descripción que puede ser usado para consultas, evolución y mantenimiento futuro de la aplicación.
- Provee múltiples vistas del mismo contenido.
- Descompone la información en páginas, navegación y presentación, que pueden ser definidas de manera independiente.
- Sitios Web que manejen gran cantidad de datos haciendo uso de interfaces dirigidas al público en general.

  Web [Ceri, 2000].

Para el desarrollo de Sistemas Web los principales modelos incluidos en la metodología de diseño WebML son:

a) Modelo de datos

b) Modelo de hipertexto

c) Modelo de presentación

2.6.1 MODELO DE DATOS

El modelo de datos representa las diferentes tablas de datos y sus relaciones que son necesarias para una aplicación Web concreta. Se pueden utilizar:

- Diagramas de Entidad-Relación (E-R) que muestran todas las tablas, los diferentes campos de cada tabla, y las relaciones entre ellas.
- Diagramas UML de clases que pueden representar la misma información que un diagrama de Entidad-Relación (por lo que puede usarse de manera equivalente), e incluso información adicional sobre el modelo de datos.

Figura 2. 5 Modelo de datos

Fuente: [www.webml.org]

2.6.2 MODELO DE HIPERTEXTO

El modelo de hipertexto especifica cómo se compone el sitio y la navegación en el sitio. La composición del sitio describe las páginas que forman parte del hipertexto y las unidades de contenido que constituyen cada página. Las páginas de un sitio Web son catalogadas como contenedores de información que es enviada al usuario. Las entidades son elementos atómicos de contenido empleadas para publicar información descrita en el modelo de datos WebML contiene (7) tipos de unidades predefinidas para desarrollar páginas Web: data, multi-data, index (y sus variantes jerárquicas y selección múltiple), entry, scroller. Cada unidad está asociada a una entidad subyacente, de la cual se obtiene el contenido. La especificación de una entidad subyacente determina el tipo de objeto del cual se deriva el contenido de la unidad.

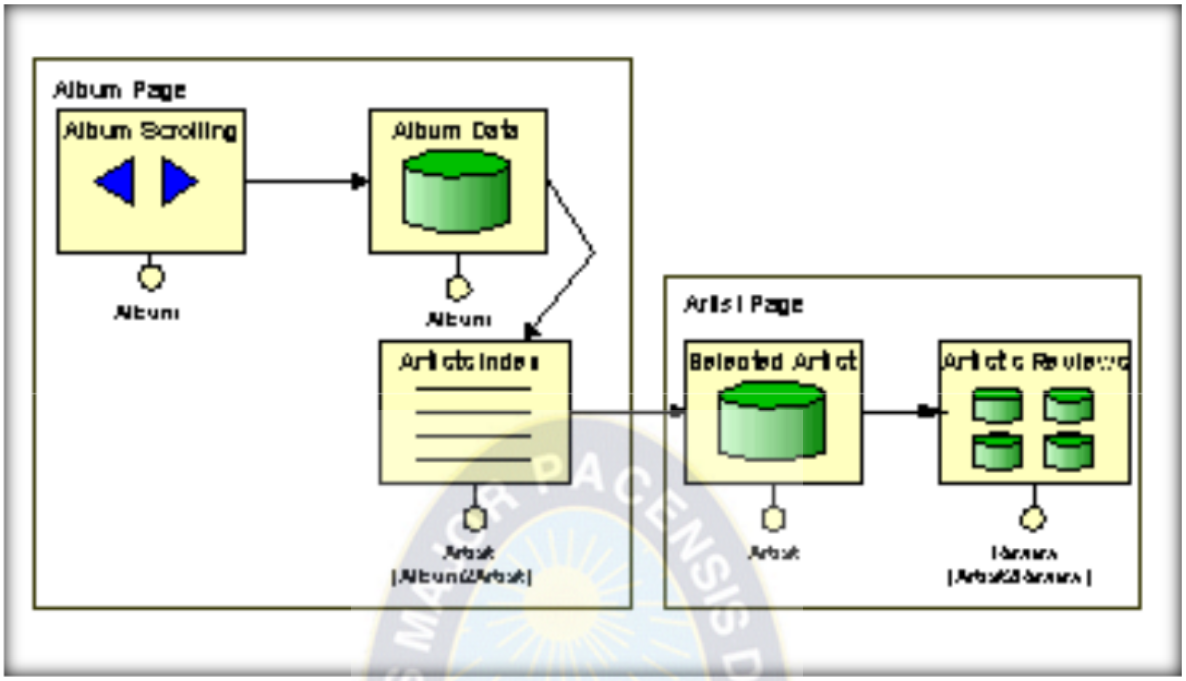


Figura 2. 6 Modelo de Hipertexto

Fuente: [www.webml.org]

2.7 TECNOLOGIA DE SOFTWARE

2.7.1 APACHE

Apache es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación. (**Fuente:** http://httpd.apache.org/ABOUT_APACHE.html)

2.7.2 PHP

PHP es un lenguaje de script (o de guiones), Es un lenguaje interpretado de alto nivel, especialmente pensado para desarrollos Web, el cual puede ser entendido en paginas HTML. La mayoría de su sintaxis es similar al lenguaje de programación “C”, “JAVA” y “PERL” que aprenderlo no es muy difícil. La meta de este lenguaje es permitir escribir a los

creadores de páginas Web, paginas dinámicas de una manera rápida y sencilla, aunque se puede hacer mucho mas con PHP, su acrónimo de PHP es: “Hipertexto Preprocessor”. (**Fuente:** http://mx.php.net/manual/es/preface.php)

2.7.3 MySQL

MySql es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU GLP, pero, empresas que quieran incorporarlo en productos propietarios puede comprar a la empresa una licencia más permisiva que les permita ese uso.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias propietarias.

MySQL fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

Las características principales de MySQL son:

- Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando

optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet. [DWP, 2000].

(**Fuente:** La Biblia MySQL Lan Gilfillan 2005)

2.7.4 AJAX

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

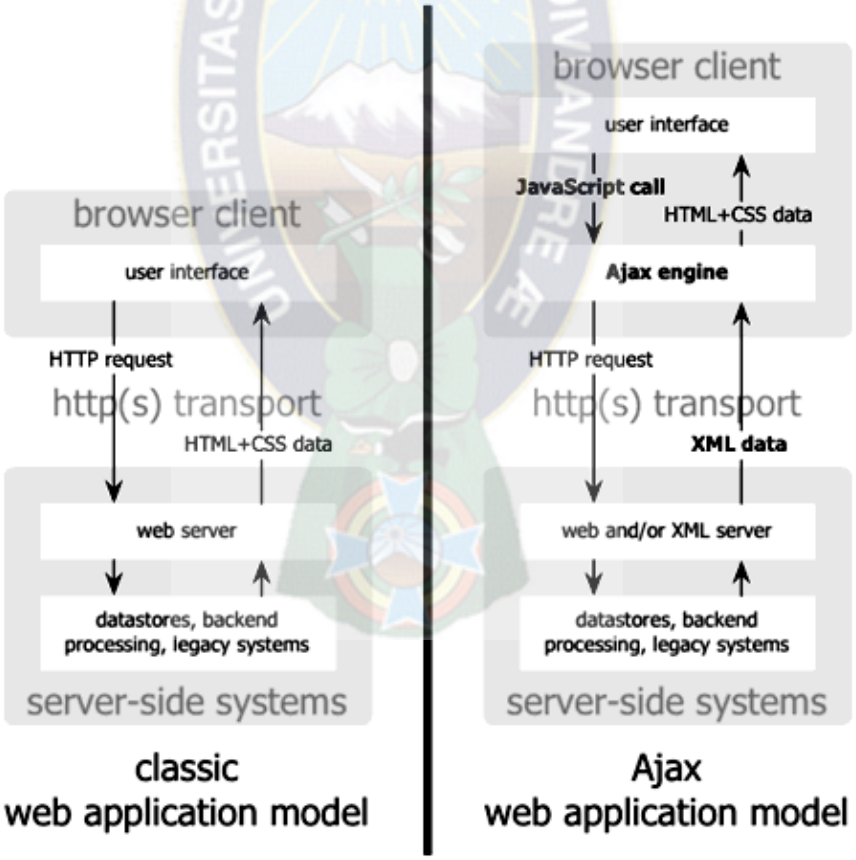


Figura 2. 7 Comparación del modelo de aplicación web clásico con Ajax

Fuente: (J2EE,2004)

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM). (**Fuente:** J2EE, 2004)

2.8 CALIDAD DE SOFTWARE

La Calidad del Software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente. La Calidad del Software (CS) es una disciplina más dentro de la Ingeniería del Software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares. Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que se sean cumplidos. (Press, 1993)

Teniendo en cuenta la definición anterior, se puede decir que los requisitos del software son la base de las medidas de calidad y que la falta de concordancia con los requisitos es una falta de calidad. Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Si no se sigue ninguna metodología siempre habrá falta de calidad. Todas las metodologías y herramientas tienen un único fin producir software de alta calidad.

2.8.1 ISO 9126

El estándar internacional ISO 9126, actualmente en proceso de revisión, ha sido el primero en definir y concretar las características de calidad que debe presentar un producto software. Según este estándar, se entiende por característica de calidad de un producto software a un conjunto de propiedades mediante las cuales se evalúa y describe su calidad.

El estándar ISO/IEC 9126 define seis características de calidad y un modelo del proceso de evaluación de productos software. Está pensado para los desarrolladores, adquirentes, personal de aseguramiento de calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software. Se utiliza para validar la completitud de una definición de requisitos, identificar requisitos software, identificar objetivos para el diseño software, identificar requisitos para las pruebas del software, identificar requisitos para el aseguramiento de la calidad e identificar criterios de aceptación para un producto software completado [GEN].

El estándar ISO 9126 - Tecnologías de la Información. Calidad de los productos software, consta de cuatro partes:

- Parte 1: Modelo de calidad (FDIS 9126-1)
- Parte 2: Métricas externas (PDTR 9126-2)
- Parte 3: Métricas internas (PDTR 9126-3)
- Parte 4: Métricas de calidad en uso (PDTR 9126-4)

FDIS indica que el documento se halla en la última fase de gestación de una Norma ISO y PDTR indica que será un informe técnico. Se explican en detalle a continuación.

a) Modelo de calidad

Un modelo de calidad es el conjunto de características y sub-características, y de cómo se relacionan entre sí. El modelo de calidad depende del tipo de producto a evaluar, ISO 9126 es un modelo genérico, por lo que es necesario adaptarlo para el tipo de producto al que se quiera aplicar. Habrá que determinar qué subcaracterísticas son relevantes, y cuáles hay que particularizar, elegir métricas específicas para el tipo de producto en cuestión y definir indicadores para las características de calidad.

El modelo de calidad del producto software de esta norma incluye calidad interna (medible a partir de las características intrínsecas, como el código fuente), calidad externa (medible en el comportamiento del producto, como en una prueba) y calidad en uso (durante la utilización efectiva por parte del usuario).

Especifica seis características para la calidad interna y externa, que se dividen en subcaracterísticas. Mientras que para la calidad en uso sólo establece cuatro características, las cuales representan el efecto combinado de las características de la calidad del software para el usuario.

La calidad de cualquier proceso del ciclo de vida del software (estándar ISO 12207) influye en la calidad del producto software, y ésta a su vez, contribuye a mejorar la calidad en el uso del producto (figura 2.7).

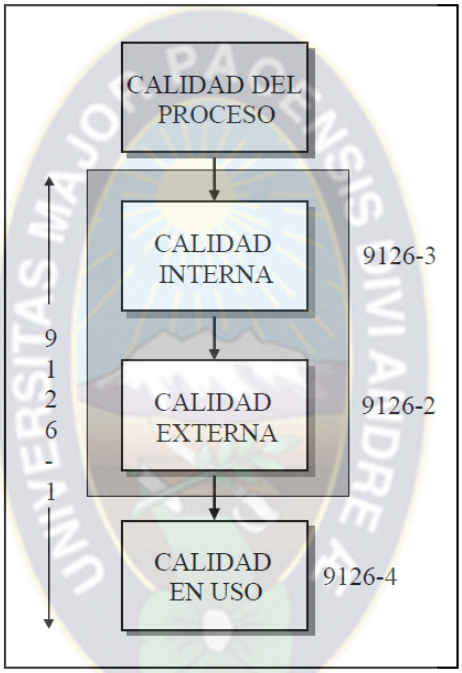


Figura 2. 8 **Calidad del producto software y estándar ISO 9126**

Fuente: Interpretación propia

La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del código cuando se ejecuta).

El modelo de calidad del estándar ISO 9126 especifica seis características (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), cada una de las cuales se divide en subcaracterísticas (figura 2.8) que se pueden medir tanto por métricas internas como externas. Son las siguientes [ISO97]:

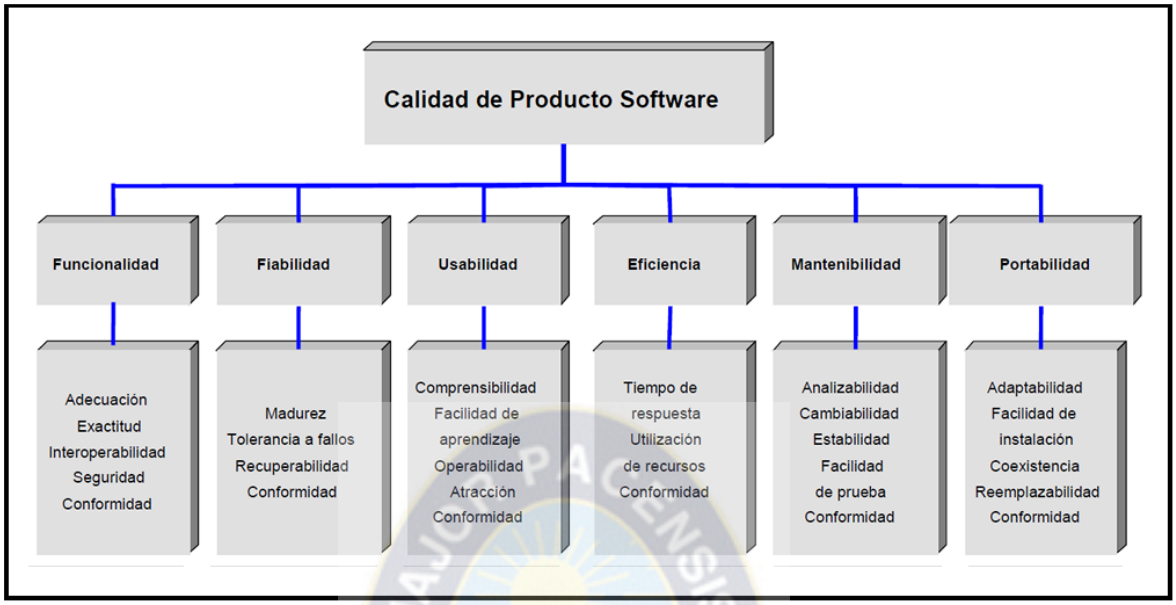


Figura 2. 9 Características y subcaracterísticas de la calidad de un producto software

Fuente: [ISO 1997]

Funcionalidad: la capacidad del producto software para proporcionar funciones que satisfagan las necesidades especificadas e implícitas cuando el software se utiliza en las condiciones especificadas.

Fiabilidad: la capacidad del producto software para mantener un nivel especificado de rendimiento cuando se utiliza en las condiciones especificadas.

Usabilidad: la capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario, cuando se utiliza en las condiciones especificadas. Depende del tipo de "uso" que se espera y tipo de "usuario" que utilizará el producto (desarrollador, evaluador/seleccionador, integrador, configurador/administrador del sistema, usuario del sistema).

Eficiencia: la capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo las especificaciones determinadas.

Mantenibilidad: la capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.

Portabilidad: la capacidad del producto software de ser transferido de un entorno a otro.

b) Selección de características, subcaracterísticas y métricas a evaluar

La selección de las características, subcaracterísticas y métricas dependerán de los objetivos del negocio para el producto y las necesidades del personal involucrado; a continuación se citan éstas necesidades:

La persona encargada de dar mantenimiento a la aplicación podría evaluar el software aplicando métricas de mantenimiento.

La persona responsable de implementar el software en diferentes ambientes podría evaluar un producto de software basado en métricas de portabilidad.

La organización que utiliza el producto de software puede usar la evaluación de la calidad para validar que los requerimientos se cumplan bajo diferentes condiciones de operación y proporcionar una retroalimentación sobre la necesidad de algún cambio a los responsables de mantenimiento.

En la Tabla 2.1 se muestra la plantilla de selección de las características y subcaracterísticas a evaluar, la que está dividida en las siguientes columnas:

Columna de características: aquí se define la característica de calidad a evaluar.

Columna de subcaracterística: Aquí se define las subcaracterísticas de calidad que vamos a evaluar.

Columna de Pesos: En esta columna se especificará el nivel de importancia de la subcaracterística dependiendo de los objetivos del negocio o del personal involucrado en el sistema.

Los pesos están dados por los valores Alto (H), Medio (M), Bajo (L) y No Aplica (NA).

El peso Alto (H), significa que el grado de importancia que tiene la subcaracteristica para el personal que es responsable del sistema es alto, por ende se realizará las mediciones.

El peso Medio (M), significa que la subcaracterística no es tan relevante para el personal responsable del sistema, y también será medida.

El peso Bajo (L), significa que la subcaracterística no tiene relevancia para el personal responsable del sistema y no será medida.

El valor (NA) se dará a las subcaracterísticas que no se pueden medir debido a factores como que la empresa no cuenta con los recursos necesarios de acuerdo a la norma ISO 9126 para medir dicha subcaracteristica, como la documentación necesaria, manuales generados, etc.

Columna Observación: En esta columna se explica por qué se da el peso a la subcaracterística. En caso de que el valor sea alto (H), ésta columna quedará en blanco.

CALIDAD EXTERNA			
CARACTERÍSTICAS	**SUBCARACTERÍSTICAS**	**PESOS (HIGH/MEDIUM/LOW/NA)**	**OBSERVACIÓN**
Funcionalidad	Adecuación		
	Exactitud		
	Interoperabilidad		
	seguridad		
Confiabilidad	Madurez (hardware/software/datos)		
	Tolerancia a errors		
	Recuperabilidad (datos, procesos, tecnología)		
Usabilidad	Entendimiento		
	Facilidad de aprendizaje		
	Operabilidad		
	Atractibilidad		
Eficiencia	Comportamiento en el tiempo		
	Utilizacion de recursos		
Mantenibilidad	Capacidad de ser analizado		
	Facilidad de cambio		
	Estabilidad		
	Facilidad de pruebas		
Portabilidad	Adaptabilidad		
	Facilidad de instalación		
	Coexistencia		
	Reemplazabilidad		

Tabla 2. 1 Ejemplo de métricas externas (Métricas de adecuación)

Fuente: ISO 9126

c) Tabla de métricas según el estándar ISO 9126

La norma ISO 9126-2 se basa en una serie de tablas en las que se encuentran definidas las métricas que corresponden a cada característica y subcaracterísticas que debe cumplir un producto de software para ser considerado de calidad.

Cada tabla contiene una serie de atributos que permiten identificar de una manera clara el objetivo de cada métrica.

A continuación se explica cada uno de los atributos expuestos en las tablas de métricas definidas por el estándar ISO 9126-2.

La siguiente información está dada para cada tabla:

Nombre: Las métricas correspondientes en la tabla para la métrica interna y la métrica externa tienen nombres similares.

Propósito: La pregunta a ser respondida con el uso de la métrica.

Método de aplicación: Proporciona un esbozo de la aplicación.

Medición, fórmula: Proporciona la fórmula de la medida y explica los elementos de los datos usados.

Interpretación: proporciona el rango y valores válidos. 71

Tipo de escala: Tipo de escala usada por la métrica. Los tipos de escala son: escala nominal, escala ordinal, escala intervalo, escala proporción y escala absoluta.

Tipo de medida: Los tipos de medida usados son: tipo tamaño (por ejemplo, tamaño de la función, tamaño de la fuente), tipo tiempo (ejemplo, tiempo transcurrido, tiempo del usuario), tipo cuenta (por ejemplo, número de cambios, número de fallos).

Fuente de medición: Fuente de datos usados en la medición.

ISO/IEC 12207 SLCP: Identifica el proceso del ciclo de vida del software donde la métrica es aplicable.

Audiencia: Identifica al usuario de los resultados de la medida.

Algunos ejemplos para la fuente de medición son: especificación de requisitos, código fuente, informe de revisión, informe de pruebas, informe de resolución de problemas, informe de mantenimiento, diseño, etc.

d) Ejemplo de tabla de métrica externa

En la tabla 2.2 se muestra un ejemplo de una métrica externa.

Métricas externas de adecuación						
Nombre de la métrica	**Propósito de la métrica**	**Método de aplicación**	**Medición, fórmula y cálculos de los elementos de dato**	**Interpretación del valor medido**	**Tipo de escala métrica**	**Tipo de medida**
Aplicación funcional de cobertura	Cuan correcta es la implementación funcional?	Hacer pruebas funcionales (prueba de la caja negra) del sistema de acuerdo a los requerimientos de especificación Contar el número de funciones implementadas incorrectamente y de las faltantes detectadas en la evaluación y compararlas con el número total de funciones descritas en los requerimientos de especificación. Contar el número de funciones que están completas contra las que no lo están.	X=1-A/B A=Número de funciones implementadas incorrectamente y de las faltantes detectadas en la evaluación. B=Número de funciones descritas en los requerimientos de especificación.	0<=X<=1 Cuanto más se acerque a 1.0 es lo mejor.	Absoluta	X=Count/Count A=Count B=Count

Tabla 2. 2 Ejemplo de métricas externas (Métricas de adecuación)

Fuente: ISO 9126

CAPITULO III

MARCO APLICATIVO

Para el desarrollo del sistema es necesario planificar los pasos bajo un marco de trabajo, con el fin de cumplir con los tiempos establecidos de entrega sin dejar de lado la calidad del sistema. En este capítulo se hace una descripción del trabajo realizado en las fases de la metodología *extreme programing* XP, en los flujos de trabajo de cada fase.

El desarrollo del Software de Seguimiento y Monitoreo de la Dirección General de Cultura del G.A.M.C. pasa por cada una de las fases de XP, con la planificación correspondiente al inicio y una evaluación de los resultados obtenidos al final de la fase.

3.1 INTERRELACION DE METODOLOGIAS

La metodología de desarrollo XP tiene cuatro fases, con actividades diferentes dentro de cada una, estas se reflejan en la tabla (ver tabla 3.1), además esta metodología nos proporciona artefactos que son la fuente de datos para el desarrollo, por su parte la metodología de modelado WebML cuenta con modelos para el diseño de Sistemas Web por lo cual dichos modelos se encuentran dentro de alguna de las actividades de XP.

Al existir diversos artefactos y modelos por parte de ambas metodologías mencionadas se seleccionó las más representativas y útiles al momento del desarrollo de este proyecto.

FASES XP	ARTEFACTOS XP	MODELOS WebML
Planificación	• Historias de usuario • Tarjeta de tarea • Plan de entregas (iteraciones)	
Diseño	• Tarjetas CRC	• Modelo estructural • Modelo de hipertexto • Modelo de

		presentación
Desarrollo	• Pantallas muertas • Metáfora del sistema •	
Pruebas	• Pruebas de aceptación	

Tabla 3. 1 Interrelación de metodologías

Fuente: (Elaboración propia)

A continuación se describirán cada una de las actividades mencionadas cada una con sus respectivos artefactos y modelos generados.

3.2 FASE DE PLANIFICACION

En esta fase se abstraerá la forma de trabajo actual mediante las historias de usuario, además de los requerimientos del *Aplicación Web*, definiremos todas las tareas que serán necesarias para poder desarrollar la *Aplicación Web* esto mediante las tarjetas de tarea y por último se realizara el *Plan de Entregas* que contendrá las iteraciones a realizar, para el desarrollo del presente Proyecto.

3.2.1 HISTORIAS DE USUARIO

A continuación se muestran las historias de usuario recolectadas de la Dirección General de Cultura del G.A.M.C., las historias fueron recabadas en distintas reuniones llevadas a cabo con el personal de dicha unidad, según el cronograma establecido.

Con las historias de usuario ilustradas a continuación se podrá conocer cómo están organizados los módulos del sistema a desarrollar y sus respectivos procesos, de acuerdo a las entrevistas realizadas se pudo establecer las necesidades plasmados en el sistema.

Historia de usuario		**Nro. 1**
Usuario: **Administrador**		**Iteración asignada:** 1
Nombre de Historia de Usuario: **Registro de usuarios nuevos**		
Prioridad: **Alta**	**Riesgo en desarrollo:** Mediano	

(Alta/Media/Baja)	
Descripciones: Se necesita registrar usuarios con privilegios para que puedan ingresar al sistema y crear, administrar un determinado evento cultural.	
Observaciones: Una vez registrado el usuario por única vez este tiene que tener permisos del administrador.	

Tabla 3. 2 Historia de Usuario Nro. 1

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 1 (ver Tabla 3.2), responde a la necesidad de tener un control de los usuarios y sus roles.

Historia de usuario		**Nro. 2**
Usuario: **Administrador**	**Iteración asignada:** 1	
Nombre de Historia de Usuario: **Control del administrador**		
Prioridad: **Alta** (Alta/Media/Baja)	**Riesgo en desarrollo:** Mediano	
Descripciones: Una vez registrados los usuarios el administrador tendrá la obligación de cambiar el estado de un usuario, es decir administrar niveles de acceso a cada usuario.		
Observaciones: Para poder dar permiso a cierta información del sistema el administrador usa un switch para poder habilitar al usuario a cierta información en el sistema.		

Tabla 3. 3 Historia de Usuario Nro. 2

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 2 (ver Tabla 3.3), responde a la necesidad del administrador de tener un control sobre los usuarios registrados y restringir al acceso algunas funcionalidades y información del sistema.

Historia de usuario		Nro. 3
Usuario: **Director – Secretaria**		**Iteración asignada:** 2
Nombre de Historia de Usuario: **Registro de comunidades**		
Prioridad: **Alta** (Alta/Media/Baja)	**Riesgo en desarrollo:** Mediano	
Descripciones: Se realiza (Altas/Bajas/Modificaciones) de comunidades con toda su información para registrar las actividades que esta realiza en el ámbito cultural, así mismo poder actualizar su información.		
Observaciones: Se gestionara la información de cada comunidad perteneciente al municipio de Copacabana para poder administrar los eventos o actividades de que realiza cada comunidad.		

Tabla 3. 4 Historia de Usuario Nro. 3

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 3 (ver Tabla 3.4), responde a la necesidad de contar con información y de poder administrar los eventos realizados por cada comunidad del municipio de Copacabana.

Historia de Usuario		Nro. 4
Usuario: **Director – Secretaria**		**Iteración asignada:** 3
Nombre de Historia de Usuario: **Registro de eventos culturales**		
Prioridad: **Alta** (Alta/Media/Baja)	**Riesgo en desarrollo:** Mediano	
Descripciones: Se realiza (Altas/Bajas/Modificaciones) de eventos indicando que comunidad la realiza, el lugar y fecha de realización, y datos propios del evento planificado. .		

Observaciones: Se podrá contar con información de cada evento con todos sus detalles.

Tabla 3. 5 Historia de Usuario Nro. 4

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 4 (ver Tabla 3.5), responde a la necesidad de tener un control de los eventos que realiza una determinada comunidad.

Historia de usuario		**Nro. 5**
Usuario: **Director – Secretaria**	**Iteración asignada:** 3	
Nombre de Historia de Usuario: **Registro de auspiciadores de un evento**		
Prioridad: **Alta** (Alta/Media/Baja)	**Riesgo en desarrollo:** Mediano	
Descripciones: Realizar (Altas/Bajas/Modificaciones) de auspiciadores o patrocinadores se reanaliza a un evento ya creado y la fecha de realización debe ser mayor a la fecha actual.		
Observaciones: Un evento puede tener uno o varios auspiciadores o patrocinadores y los cuales deben estar registrados en el sistema.		

Tabla 3. 6 Historia de Usuario Nro. 5

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 5 (ver Tabla 3.6), responde a la necesidad de tener un control de los auspiciadores de un determinado evento y los patrocinadores de los mismos.

Historia de usuario		**Nro. 6**
Usuario: **Director – Secretaria**	**Iteración asignada:** 4	
Nombre de Historia de Usuario: **Búsqueda**		
Prioridad: **Alta**	**Riesgo en desarrollo:** Mediano	

(Alta/Media/Baja)	
Descripciones: Se tendrá un buscador el cual solucione problemas de tiempo que de acuerdo a opciones de búsqueda el usuario podrá tener acceso más ágil.	
Observaciones: Se podrá tener un informe detallado de la búsqueda que desee.	

Tabla 3. 7 Historia de Usuario Nro. 6

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 6 (ver Tabla 3.7), responde a la necesidad de tener acceso rápido a información de un determinado evento en base a criterios de búsqueda (fecha, nombre).

Historia de usuario		**Nro. 7**
Usuario: **Administrador – Director – Secretaria**	**Iteración asignada:** 4	
Nombre de Historia de Usuario: **Reportes**		
Prioridad: **Alta** (Alta/Media/Baja)	**Riesgo en desarrollo:** Mediano	
Descripciones: Se tendrá la opción de realizar reportes cuando el (los) usuarios autorizados así lo requieran ya que es de suma importancia tener un control sobre todos los eventos a realizarse.		
Observaciones: Sin observaciones.		

Tabla 3. 8 Historia de Usuario Nro. 7

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 7 (ver Tabla 3.8), responde a la necesidad de tener acceso a reportes por día, mes, año de los eventos que son realizados por la Dirección General de Culturará.

Historia de usuario		Nro. 8
Usuario: **Oficial Mayor de Cultura**	**Iteración asignada:** 5	
Nombre de Historia de Usuario: **Indicadores y estadísticas**		
Prioridad: **Alta** (Alta/Media/Baja)	**Riesgo en desarrollo:** Mediano	
Descripciones: Se requieren indicadores como ser graficas estadísticas que permitan visualizar porcentajes de participación y ejecución de eventos por cada comunidad.		
Observaciones: Se puede utilizar graficas como ser tortas, barras o alguna otra herramienta		

Tabla 3. 9 Historia de Usuario Nro. 8

Fuente: [Elaboración propia]

HISTORIA DE USUARIO Nro. 8 (ver Tabla 3.9), responde a la necesidad ejecutiva de poder visualizar mediante indicadores gráficos y estadísticos el estado de los eventos en ejecución.

3.2.2 TASK CART (TARJETA DE TAREA)

A continuación se describen todas las tarjetas de tarea que corresponderían a cada una de las historias de usuario descritas anteriormente.

		TARJETA DE TAREA
Numero de Tarea: 1.1	**Historia de usuario:** 1. Registro de usuarios nuevos	
Nombre de tarea: desarrollar el módulo de registro de usuarios		
Tipo de tarea: Desarrollo		
Programador responsable: Peter Paul Castillo valencia		
Descripción: realizar un módulo de registro de usuarios donde se obtenga un formulario el en que se pueda registrar, el modulo debe poder albergar los datos necesarios de cada usuario.		

Tabla 3. 10 Tarjeta de tarea desarrollo Modulo de registro de usuarios

Fuente: [Elaboración propia]

Tarjeta de tarea 1.1 (ver Tabla 3.10), describe la tarea a realizar al momento de desarrollar el módulo de registro de usuarios, así como la funcionalidad que esta deba tener para lograr su objetivo.

	TARJETA DE TAREA
Numero de Tarea: 1.2	**Historia de usuario: 1.** Registro de usuarios nuevos
Nombre de tarea: Implementación de la base de datos	
Tipo de tarea: implementación	
Programador responsable: Peter Paul Castillo valencia	
Descripción: en función al modelo estructural se implementara la base de datos en MySQL y para consultas se realizara como SQL	

Tabla 3. 11 Tarjeta de Tarea Implementación de la base de datos

Fuente: [Elaboración propia]

Tarjeta de Tarea 1.2 (ver Tabla 3.11), describe como se implementara la base de datos y que características tendrá para su correcto funcionamiento.

	TARJETA DE TAREA
Numero de Tarea: 1.3	**Historia de usuario:** 1. Registro de usuarios nuevos
Nombre de tarea: Desarrollo de la interfaz	
Tipo de tarea: Implementación	
Programador responsable: Peter Paul Castillo valencia	
Descripción: se desarrollara la interfaz de acuerdo a la facilidad de trabajo para los encargados así como las necesidades que requieran como colores de la empresa, diseños de logotipos y formulario de presentación.	

Tabla 3. 12 Tarjeta de tarea: Desarrollo de la interfaz

Fuente: [Elaboración propia]

Tarjeta de Tarea 1.3 (ver Tabla 3.12), describe la forma en la que se implementara la interfaz de usuario para el módulo de Registro de usuarios nuevos previa coordinación con cliente.

TARJETA DE TAREA	
Numero de Tarea: 1.4	**Historia de usuario:** 1. Registro de usuarios nuevos
Nombre de tarea: Reporte de usuarios	
Tipo de tarea: Mejora	
Programador responsable: Peter Paul Castillo valencia	
Descripción: el módulo de registro de usuarios dispone de un informe sobre un resumen en general de toda la información almacenada por cada usuario y sus actividades.	

Tabla 3. 13 Tarjeta de Tarea Reporte de usuarios

Fuente: [Elaboración propia]

Tarjeta de Tarea 1.3 (ver Tabla 3.13), describe la forma de representar los reportes de los usuarios y sus respectivas actividades realizadas.

TARJETA DE TAREA	
Numero de Tarea: 2.1	Historia de usuario: 2. Control del administrador
Nombre de tarea: Desarrollar módulo de Administrador	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Desarrollar un módulo en el que el administrador podrá controlar cambios de altas bajas y modificaciones, en la que este podrá tener un control de los datos de los usuarios.	

Tabla 3. 14 Tarjeta de tarea Desarrollar módulo de administrador

Fuente: [Elaboración propia]

Tarjeta de Tarea 2.1 (ver Tabla 3.14), describe la forma de desarrollar el módulo de administrador y las funcionalidades que debe cumplir.

TARJETA DE TAREA	
Numero de Tarea: 2.2	Historia de usuario: 2. Control del administrador
Nombre de tarea: Modulo configuración del sistema	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Desarrollar un módulo que nos permita administrar la configuración del sistema.	

Tabla 3. 15 Modulo configuración del sistema

Fuente: [Elaboración propia]

Tarjeta de Tarea 2.2 (ver Tabla 3.15), describe la una de las funcionalidades del administrador del sistema.

TARJETA DE TAREA	
Numero de Tarea: 2.3	Historia de usuario: 2. Control del administrador
Nombre de tarea: Implementación de la base de datos	
Tipo de tarea: implementación	
Programador responsable: Peter Paul Castillo valencia	
Descripción: En función al modelo estructural se implementara la base de datos relacionados con el módulo de administración.	

Tabla 3. 16 Implementación de tablas modulo administrador

Fuente: [Elaboración propia]

Tarjeta de Tarea 2.3 (ver Tabla 3.16), implementar las tablas necesarias en función al modelo estructural de datos relacionados con el módulo de administración.

TARJETA DE TAREA	
Numero de Tarea: 2.4	Historia de usuario: 1. Control del administrador
Nombre de tarea: Desarrollo de la interfaz de administrador	
Tipo de tarea: Implementación	
Programador responsable: Peter Paul Castillo valencia	
Descripción: se desarrollara la interfaz intuitiva para facilitar el trabajo del administrador al momento de realizar Altas, Modificaciones, Bajas..	

Tabla 3. 17 Desarrollo de la interfaz de administrador

Fuente: [Elaboración propia]

Tarjeta de Tarea 2.4 (ver Tabla 3.17), describe la forma en la que se implementara la interfaz de administrador para el módulo de Administración.

TARJETA DE TAREA	
Numero de Tarea: 2.5	Historia de usuario: 2. Control del administrador
Nombre de tarea: Utilización de JQuery	
Tipo de tarea: Mejora	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Se implementara código JQuery para las validaciones y búsquedas y procesos asíncronos para una mejor experiencia del usuario.	

Tabla 3. 18 Utilización de JQuery modulo administrador

Fuente: [Elaboración propia]

Tarjeta de Tarea 2.5 (ver Tabla 3.18), describe como utilizara la librería JQuery para realizar las validaciones y solicitudes asíncronas para una experiencia más agradable del usuario final.

TARJETA DE TAREA	
Numero de Tarea: 3.1	Historia de usuario: 3. Registro de comunidades
Nombre de tarea: Modulo registro de comunidades	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: se desarrollara un módulo de registro con (Altas, Modificaciones, Bajas) de comunidades pertenecientes al municipio de Copacabana, registrando datos referenciales y necesarios.	

Tabla 3. 19 Modulo Registro de comunidades

Fuente: [Elaboración propia]

Tarjeta de Tarea 3.1 (ver Tabla 3.19), describe cómo se va desarrollar el módulo de registro de comunidades y las funciones básicas que debería presentar..

TARJETA DE TAREA	
Numero de Tarea: 3.2	Historia de usuario: 3. Registro de comunidades
Nombre de tarea: Interfaz de usuario modulo registro de comunidades	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: se implementara una interfaz de usuario con formularios ordenados con los campos necesarios para una interacción con el usuario de una forma amigable y con las opciones necesarias para realizar (CRUD).	

Tabla 3. 20 Tarjeta de Tarea: Interfaz de usuario modulo registro de comunidades

Fuente: [Elaboración propia]

Tarjeta de Tarea 3.2 (ver Tabla 3.20), describe las características de la interfaz de usuario para el módulo de registro de comunidades y las funcionalidades básicas que debería proveer.

TARJETA DE TAREA	
Numero de Tarea: 4.1	Historia de usuario: 4. Registro de eventos culturales
Nombre de tarea: Programar las clases para el Registro de eventos culturales	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Crear las clases necesarias para el módulo de registro de eventos culturales en el módulo de Registro de Comunidades donde se tenga la opción de realizar un CRUD de eventos en una determinada comunidad.	

Tabla 3. 21 Tarjeta de Tarea: Desarrollo sub modulo Registro de eventos culturales
Fuente: [Elaboración propia]

Tarjeta de Tarea 4.1 (ver Tabla 3.21), describe las características para desarrollar un sub módulo de registro de eventos culturales en el módulo de Registro de comunidades con las funciones necesarias para realizar un CRUD asociado a una comunidad.

TARJETA DE TAREA	
Numero de Tarea: 4.2	Historia de usuario: 4. Registro de eventos culturales
Nombre de tarea: Desarrollo Interfaz sub modulo Registro de eventos culturales	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Desarrollar una interfaz en la que nos permita seleccionar una comunidad y nos proporcione un formulario debidamente validado en el que se pueda registrar las características e información del evento que se desea crear.	

Tabla 3. 22 Tarjeta de Tarea: Desarrollo Interfaz sub modulo Registro de eventos culturales
Fuente: [Elaboración propia]

Tarjeta de Tarea 4.2 (ver Tabla 3.22), describe las características que debería tener la interfaz para el usuario en el sub módulo de registro de eventos de cultura.

TARJETA DE TAREA	
Numero de Tarea: 5.1	Historia de usuario: 5. Registro de auspiciadores de un evento
Nombre de tarea: Desarrollo sub modulo Registro de auspiciadores	
Tipo de tarea: Desarrollo	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Desarrollar un sub módulo de registro de auspiciadores de evento dentro del módulo de registro de eventos de cultura.	

Tabla 3. 23 Tarjeta de Tarea: Desarrollo sub modulo Registro de auspiciadores

Fuente: [Elaboración propia]

Tarjeta de Tarea 5.1 (ver Tabla 3.23), describe las características que se debe desarrollar par el sub módulo de registro de auspiciadores.

TARJETA DE TAREA	
Numero de Tarea: 6.1	Historia de usuario: 6. Búsqueda
Nombre de tarea: Implementar herramientas de búsqueda	
Tipo de tarea: Mejora	
Programador responsable: Peter Paul Castillo valencia	
Descripción: implementar herramientas de búsqueda bajo criterios como ser fecha, relevancia, por comunidad.	

Tabla 3. 24 Tarjeta de Tarea: Implementar herramientas de búsqueda

Fuente: [Elaboración propia]

Tarjeta de Tarea 6.1 (ver Tabla 3.24), describe las características para implementar una herramienta de búsqueda.

TARJETA DE TAREA	
Numero de Tarea: 7.1	Historia de usuario: 7. Reportes
Nombre de tarea: Desarrollar módulo de reportes	
Tipo de tarea: Desarrollo	

Programador responsable: Peter Paul Castillo valencia
Descripción: Desarrollar un módulo de reportes el cual permita al usuario obtener información de las actividades de las comunidades y sus eventos, y estos se puedan generar en archivos digitales.

Tabla 3. 25 Tarjeta de Tarea: Desarrollar módulo de reportes

Fuente: [Elaboración propia]

Tarjeta de Tarea 7.1 (ver Tabla 3.25), describe las características para el desarrollo de un módulo de reportes con niveles de acceso de acuerdo al rol del usuario.

	TARJETA DE TAREA
Numero de Tarea: 8.1	Historia de usuario: 8. Indicadores y estadísticas
Nombre de tarea: Implementar herramientas estadísticas	
Tipo de tarea: Implementar	
Programador responsable: Peter Paul Castillo valencia	
Descripción: Implementar herramientas graficas de indicadores en base a la información almacenada en la base de datos ya sea como graficas (tortas, barras, porcentajes) que permita al usuario monitorear y poder tener una idea clara de el avance y ejecución de los eventos programados o eventos pasados para nuevas proyecciones.	

Tabla 3. 26 Tarjeta de Tarea: Implementar herramientas estadísticas

Fuente: [Elaboración propia]

Tarjeta de Tarea 8.1 (ver Tabla 3.26), describe las características para la implementación de indicadores estadísticos para el seguimiento y monitoreo de eventos que están registrados en el sistema.

3.2.3 PLAN DE ENTREGAS

En el plan de entregas se dará a conocer el número de iteraciones en el cual una o varias historias de usuario serán producidas, se tomó en cuenta todas las observaciones que hizo el oficial mayor de cultura. De esta manera el plan de entregas será descrito a continuación.

Nro.	Tarea	Fecha		2015			
		Inicio	Fin	02	03	04	05
1	Tarea: 1.1 Tarea: 1.2 Tarea: 1.3 Tarea: 1.4	10/02/2015	30/02/2015				
2	Tarea: 2.1 Tarea: 2.2 Tarea: 2.3 Tarea: 2.4 Tarea: 2.5	02/03/2015	20/03/2015				
3	Tarea: 3.1 Tarea: 3.2 Tarea: 4.1 Tarea: 4.2 Tarea: 5.1	23/03/2015	10/04/2015				
4	Tarea: 6.1 Tarea: 7.1	13/04/2014	02/05/2015				
5	Tarea: 8.1	04/05/2015	29/05/2015				

Tabla 3. 27 Plan de entregas

Fuente: [Elaboración propia]

3.2.4 VELOCIDAD DEL PROYECTO

Como pudimos observar en la tabla 3.27 contaremos con 5 iteraciones, es así que dividiremos el tiempo que tenemos para desarrollar el sistema, aproximadamente 4 meses, en tiempos de iteración relativamente iguales. A fin de no acelerar ni ralentizar el proyecto y que el mismo se desarrolle de manera prácticamente continua y sin mucha presión.

También podemos promediar la velocidad del proyecto utilizando

Velocidad del proyecto = Numero de historias de usuario/Numero de iteraciones

Velocidad del proyecto = 8/5

Velocidad del Proyecto = 1.6

Este indicador se interpreta de la siguiente manera: en cada iteración se avanzara en promedio 1 historias de usuario y un 60% de la historia de usuario siguiente.

3.2.5 PLANIFICACION DE ITERACIONES

De acuerdo a las distintas historias de usuario y las estimaciones que hicimos en la fase de planificación tenemos la siguiente tabla de planificación de iteraciones:

	Febrero				Marzo				Abril				Mayo			
eración	1° semana	2° semana	3° semana	4° semana	5° semana	6° semana	7° semana	8° semana	9° semana	10°	11°	12°	13°	14°	15°	16°
1																
2																
3																
4																
5																

Tabla 3. 28 Planificación de Iteraciones

Fuente: [Elaboración propia]

3.2.6 METAFORA DEL SISTEMA

La metáfora del sistema nos permitirá una comunicación entre el cliente y el equipo de desarrollo. En si el objetivo de la misma es el poder llegar a obtener un lenguaje común para ambas partes.

METAFORA DEL SISTEMA
Todo evento o actividad realizada que tenga como objeto proteger, fomentar y promocionar el dialogo intercultural y desarrollo cultural y creativo en el municipio de Copacabana, está relacionado con actividades que involucran a la dirección general de cultura del G.A.M.C

Tabla 3. 29 Metáfora del sistema

Fuente: [Elaboración propia]

3.3 DISEÑO

Para el diseño se tomara en cuenta los siguientes puntos.

3.3.1 TARJETAS CRC

la finalidad de estas tarjetas es facilitar la implementación de las clases definidas anteriormente, las mismas que serán utilizadas en el desarrollo del Software orientado a la Web, a continuación se presentan todas las tarjetas C.R.C.

USUARIOS	
Objetivos	**Colaboradores**
Alta de usuario **Modificación de usuario** **Eliminar usuario** **Recuperar contraseña**	Jquery Validate, para validar los campos de ingreso de información Repositorio Usuarios Administrador del sistema Listado de roles

Tabla 3. 30 Tarjeta C.R.C. Usuarios.

Fuente: [Elaboración propia]

PATROCINADORES	
Objetivos	**Colaboradores**
Alta de auspiciador **Modificación de auspiciador** **Eliminar auspiciador** **Listado de auspiciadores**	• Jquery Validate, para validar los campos de ingreso de información • Repositorio auspiciadores • Usuario autentificado con privilegios

Tabla 3. 31 Tarjeta C.R.C. Auspiciadores.

Fuente: [Elaboración propia]

COMUNIDADES	
Objetivos	**Colaboradores**
Alta de comunidad **Modificación de registros de comunidad** **Eliminar comunidad**	• Jquery Validate, para validar los campos de ingreso de información. • Repositorio Comunidades. • Usuarios registrados en el sistema. Autentificado

Tabla 3. 32 Tarjeta C.R.C. Comunidades.

Fuente: [Elaboración propia]

EVENTOS	
Objetivos	**Colaboradores**
Alta de eventos **Modificación de registros de evento** **Agregar auspiciadores** **Eliminar evento** **Asignar comunidad** **Registro de fecha** **Listar eventos**	• Jquery Validate, para validar los campos de ingreso de información. • Repositorio eventos • Repositorio comunidades • Repositorio tipo eventos • Usuarios registrados en el sistema. Autentificado

Búsqueda de eventos **Vista de registro evento**	

Tabla 3. 33 Tarjeta C.R.C. Eventos.

Fuente: [Elaboración propia]

REPORTES	
Objetivos	**Colaboradores**
Reporte por comunidades **Reporte por fechas** **Reporte por auspiciador** **Generar datos formato JSON** **Exportar reportes** **Opciones de exportación en archivos digitales**	• Repositorio eventos • Repositorio comunidades • Repositorio eventos • Repositorio usuarios • Repositorio auspiciadores • Repositorio tipo eventos • Highcharts API para graficar los datos. • Jquery Framework como dependencia de Highcharts API.

Tabla 3. 34 Tarjeta C.R.C. Reportes.

Fuente: [Elaboración propia]

3.3.2 MODELO DE DATOS

El modelo estructural (ver figura 3.1) representara el total de las entidades y relaciones a utilizar en el presente proyecto estos se ven representados en forma de tablas.

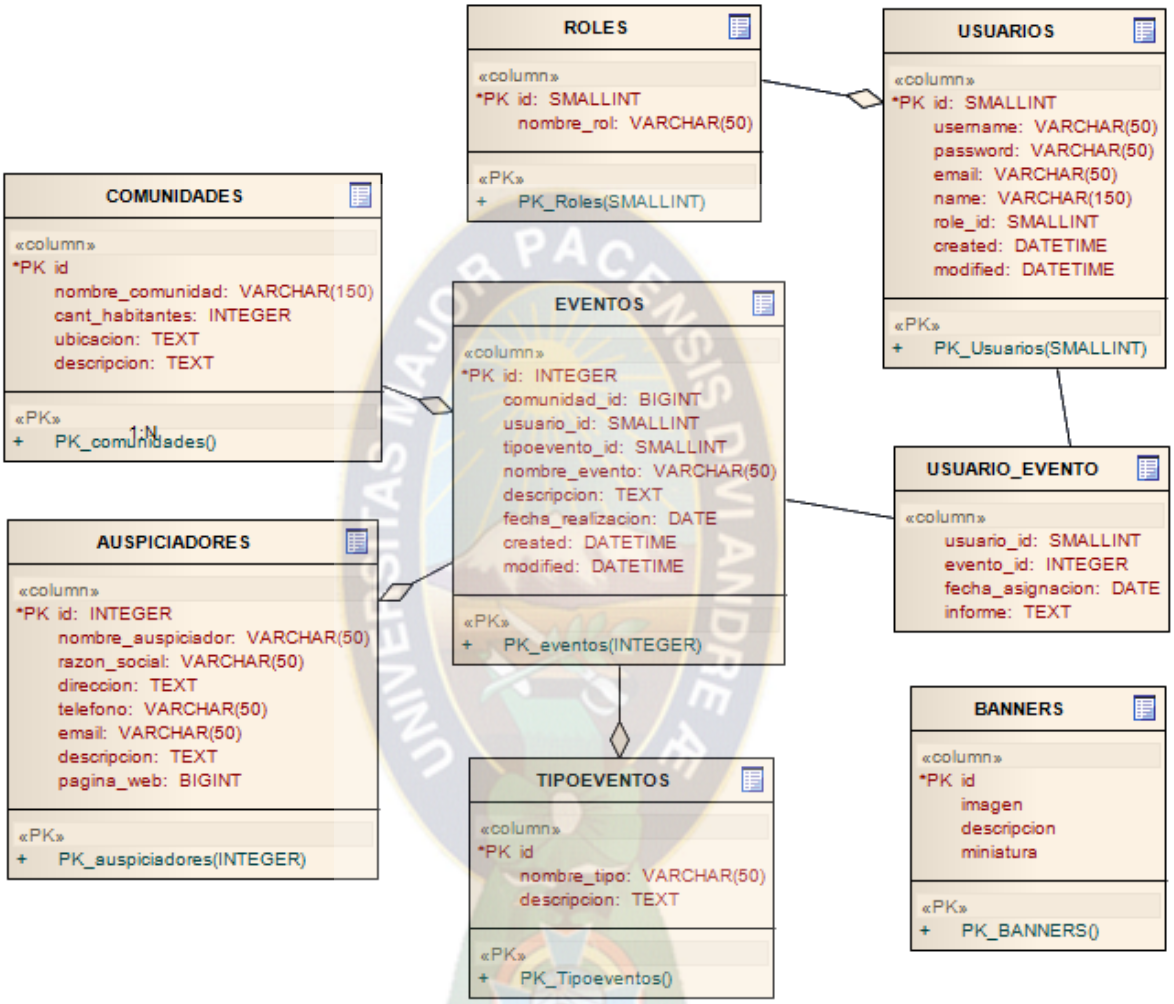


Figura 3. 1 Modelo de Datos

Fuente: [Elaboración propia]

El modelo físico (ver figura 3.2) representara el total de las entidades y relaciones y tipos de datos y longitudes como irán definidas en el gestor de base de datos en nuestro caso MySQL.

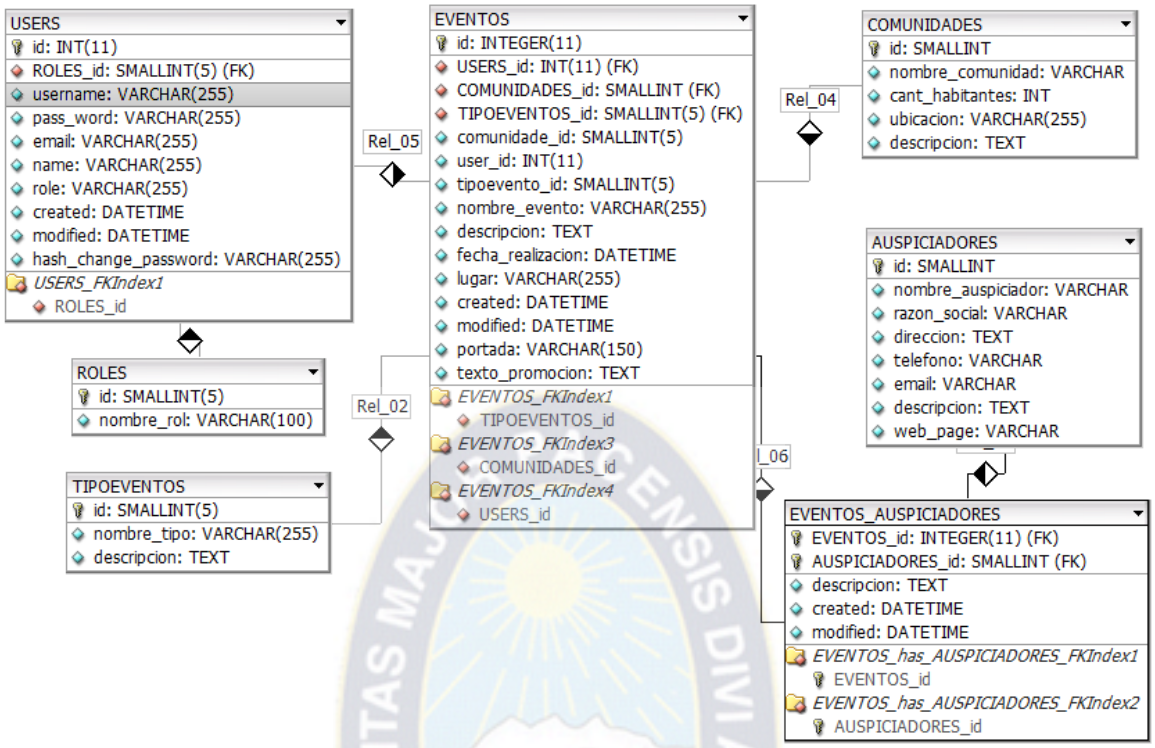


Figura 3. 2 Modelo físico de datos

Fuente: [Elaboración propia]

3.3.3 PRIMERA ITERACIÓN

Uno de los requisitos de la metodología WebML incluye lo que son modelos de presentación de los diagramas de hipertexto en forma visual, de esta manera a continuación se muestran dichos modelos para su posterior implementación y desarrollo

3.3.3.1 Modelo de Hipertexto

El diagrama (ver Figura 3.2) se muestra la forma en la que se va desarrollar el módulo de registro de usuarios nuevos que pertenecen a la historia de usuario 1 y a sus respectivas Tareas. En este se muestra la parte navegación de este módulo.

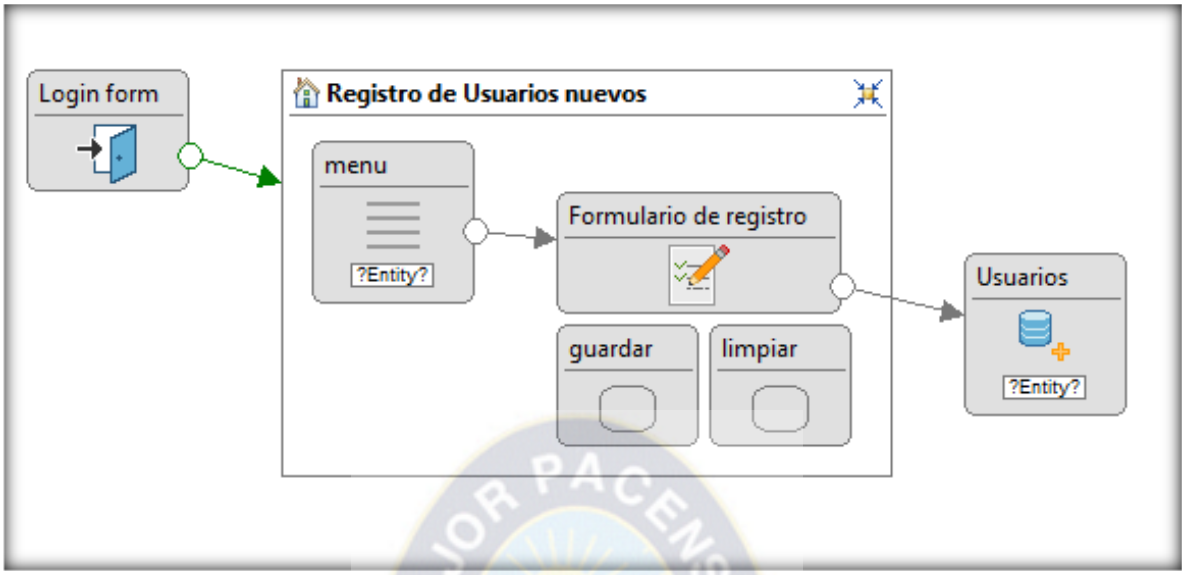


Figura 3. 3 Diagrama de hipertexto: Registro de usuarios

Fuente: [Elaboración propia]

En el siguiente diagrama (ver Figura 3.2) se muestra la manera navegación de la tarea 1.3 correspondiente a la historia de usuario nro. 1 en el que se representa el acceso a la información de los usuarios y sus respectivas actividades.

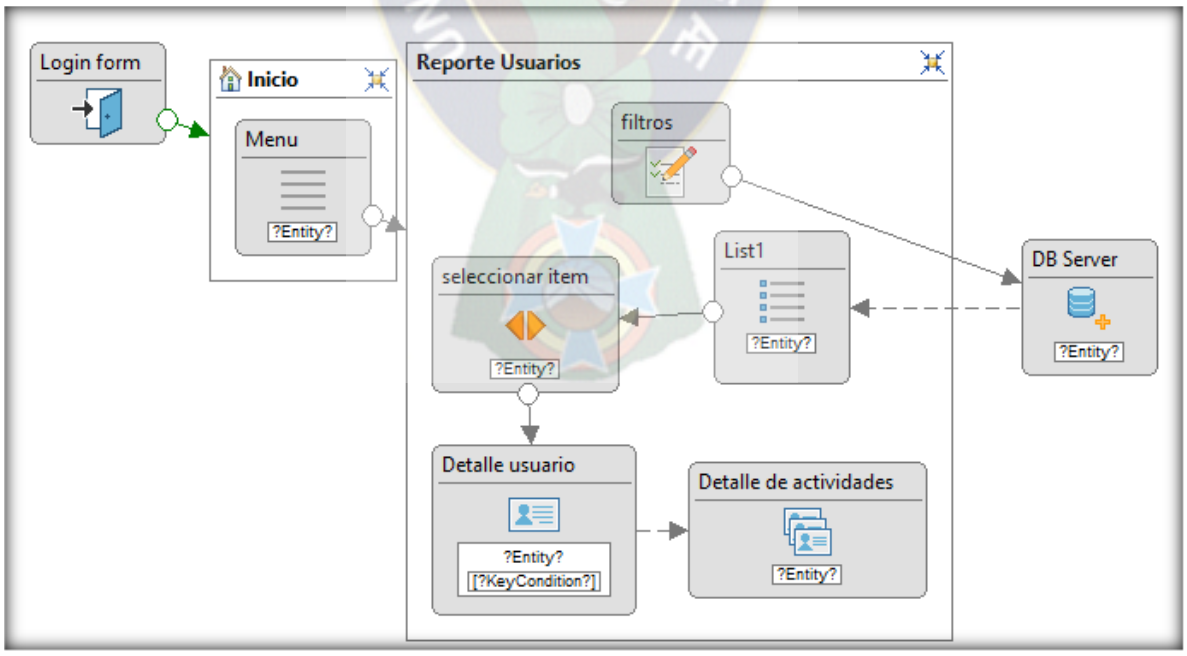


Figura 3. 4 Diagrama de hipertexto: Reportes de usuarios

Fuente: [Elaboración propia]

3.3.3.2 Modelo de Presentación

La pantalla autentificación de usuario (ver figura 3.4) se muestra el formulario de registro de autentificación de usuarios.

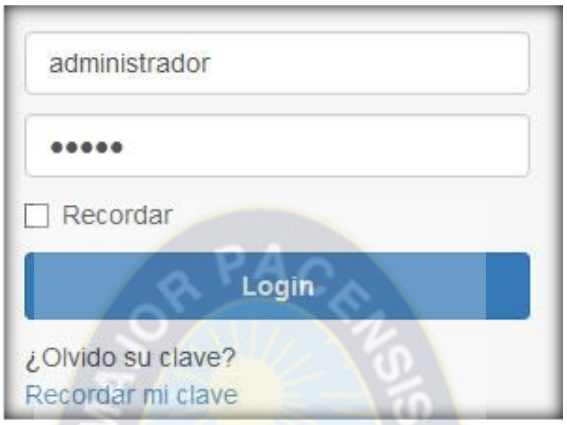


Figura 3. 5 Diagrama de autentificación de usuario

Fuente: [Elaboración propia]

La pantalla registro de usuario (ver figura 3.5) se muestra el formulario con los campos requeridos para el registro de usuarios nuevos.

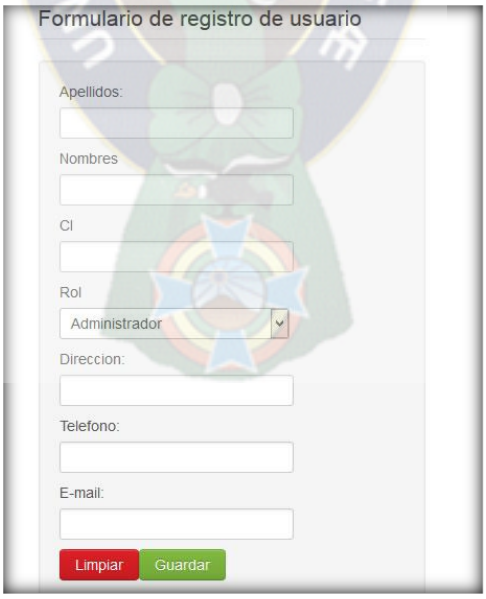


Figura 3. 6 Diagrama Registro de Usuarios nuevos

Fuente: [Elaboración propia]

3.3.4 SEGUNDA ITERACIÓN

A continuación se describen los diagramas de hipertexto y diagramas de presentación para las historias de usuarios que corresponden a la segunda iteración.

3.3.4.1 Modelo de Hipertexto

En la siguiente diagrama (Ver Figura: 3.6)se muestra la estructura del proceso de administración de usuarios del sistema.

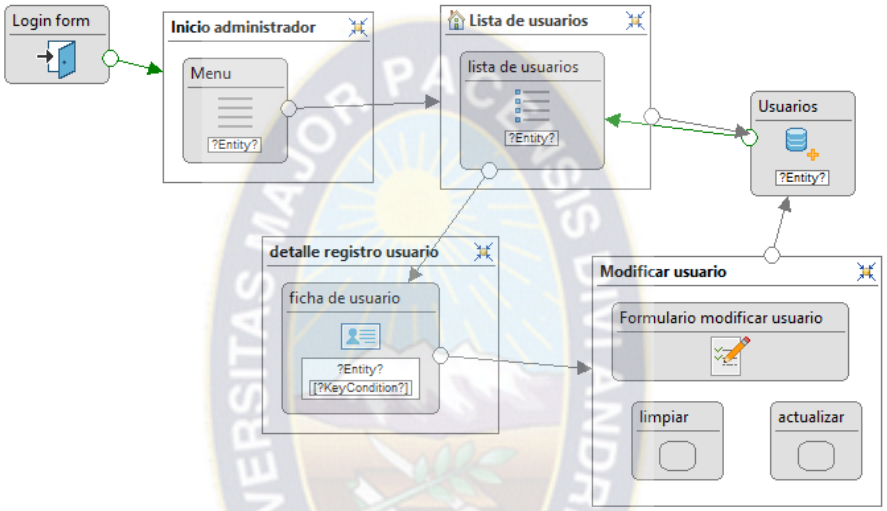


Figura 3. 7 Diagrama de hipertexto: Administración de usuarios.

Fuente: [Elaboración propia]

3.3.4.2 Modelo de Presentación

La pantalla Usuarios muestra la lista de todos los usuarios registrados en el sistema (ver Figura 3.7) , ya que usuarios pueden cumplir un determinado rol tenemos las opciones de (Eliminar/Modificar)

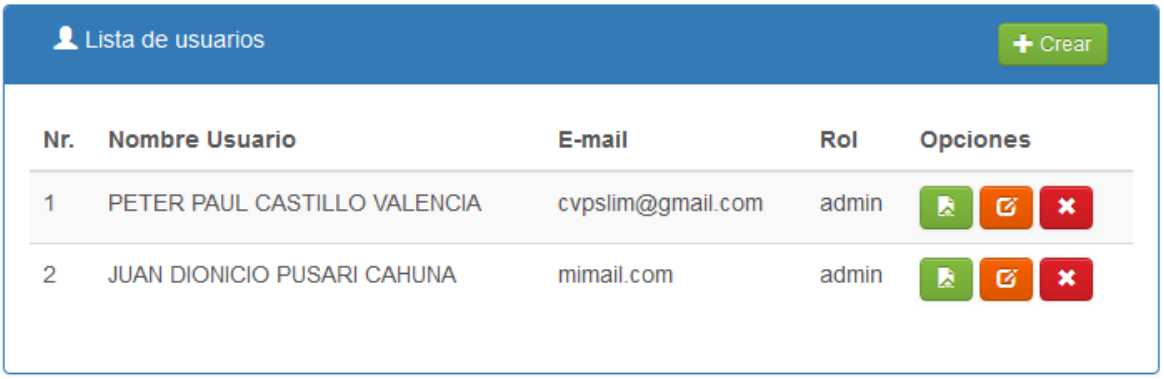


Figura 3. 8 Diagrama lista de usuarios registrados.

Fuente: [Elaboración propia]

En la figura (ver Figura: 3.8) se muestra el detalle de registro del usuario para la verificación de datos o información del administrador.

Figura 3. 9 Diagrama ficha registro de usuario.

Fuente: [Elaboración propia]

3.3.5 TERCERA ITERACIÓN

A continuación se describen los diagramas de hipertexto y diagramas de presentación para las historias de usuarios que corresponden a la tercera iteración.

3.3.5.1 Modelo de Hipertexto

En la siguiente diagrama (Ver Figura 3.9), se muestra la estructura del proceso de registro de comunidades corresponde a la historia de usuario Nro. 3 y sus respectivas tareas.

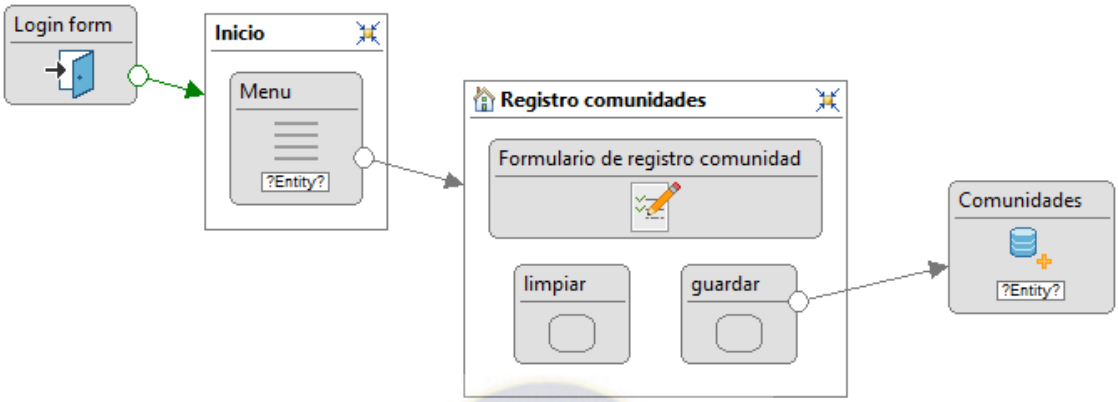


Figura 3. 10 Modelo de hipertexto: Registro de comunidades

Fuente: [Elaboración propia]

En la siguiente diagrama (Ver Figura 3.10) se muestra la estructura del proceso de actualización de registro de comunidades corresponde a la historia de usuario Nro. 3

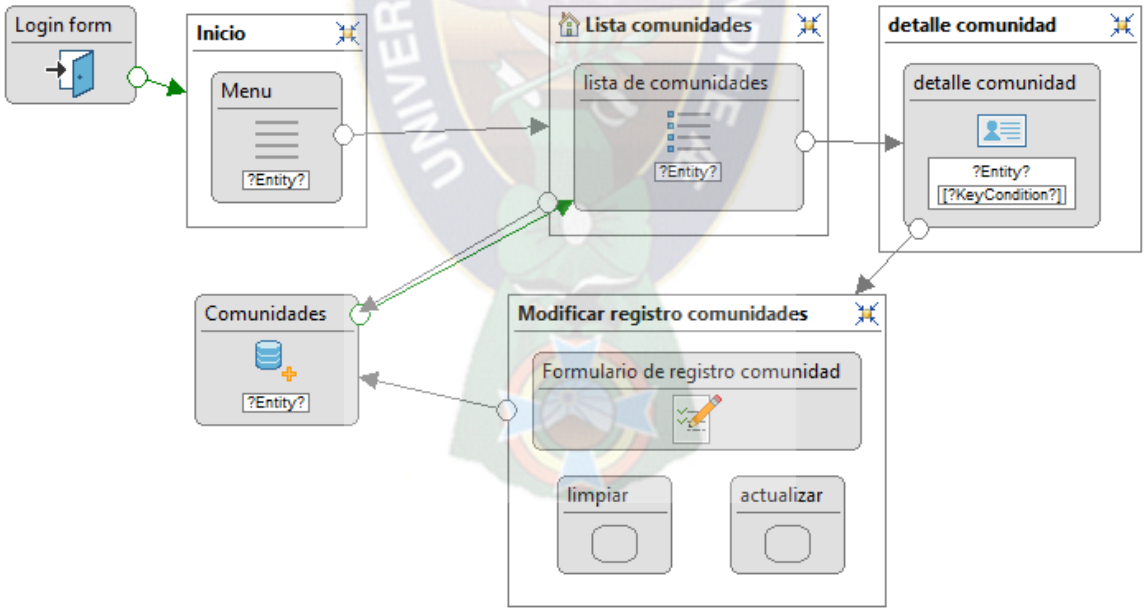


Figura 3. 11 Modelo de hipertexto: Modificar comunidades.

Fuente: [Elaboración propia]

En el siguiente diagrama (Ver Figura 3.11) se muestra la estructura para la creación de nuevos eventos.

Figura 3. 12 Modelo de hipertexto: Creación de eventos.

Fuente: [Elaboración propia]

El siguiente diagrama (Ver Figura 3.12) describe la estructura para ver detalle de evento.

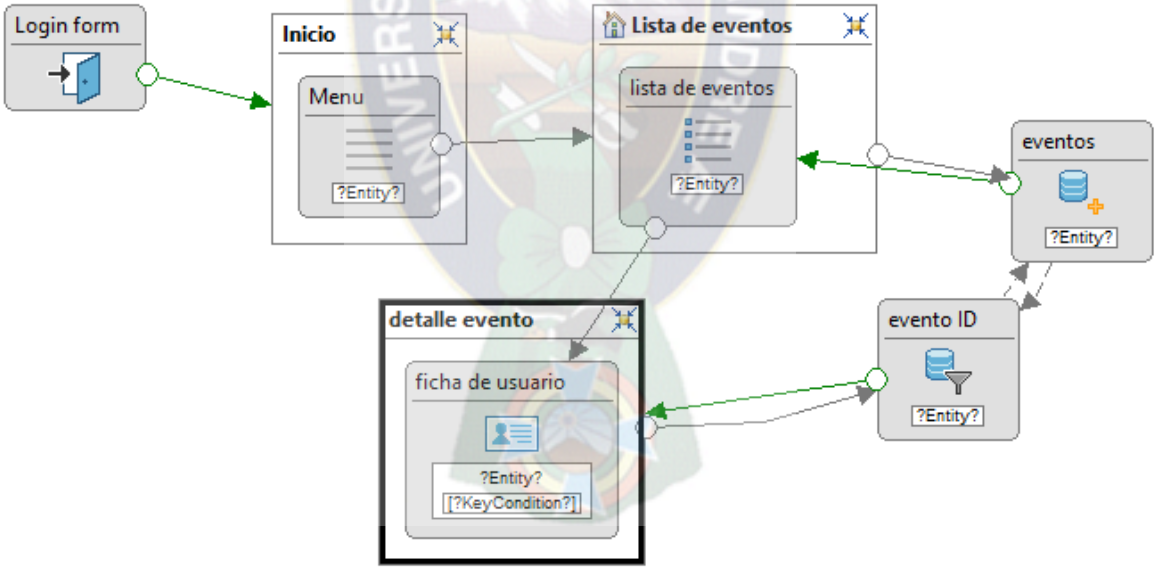


Figura 3. 13 Modelo de hipertexto: Detalle de evento.

Fuente: [Elaboración propia]

En el siguiente diagrama (Ver Figura 3.13) se describe la asignación de auspiciadores para cada evento ya que en la creación no se tendrá inicialmente auspiciadores se le asigna en la opción de modificar evento.

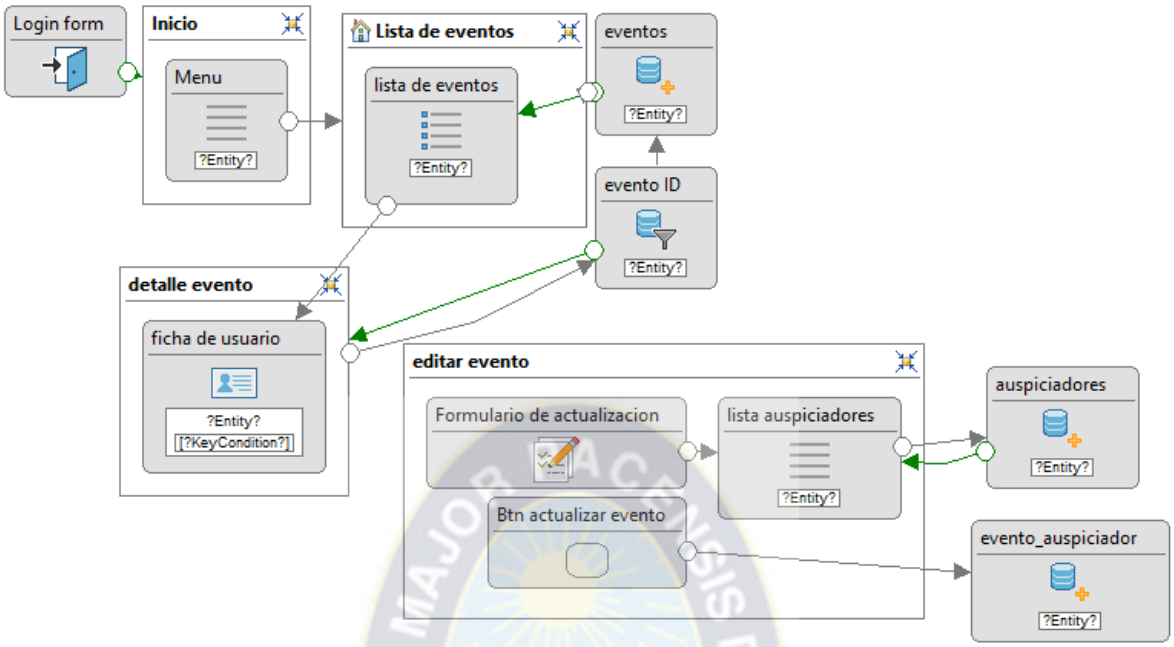


Figura 3. 14 Modelo de hipertexto: Asignar auspiciador evento.

Fuente: [Elaboración propia]

3.3.5.2 Modelo de Presentación

La pantalla Comunidades muestra la lista de todas las comunidades registradas en el sistema (ver Figura 3.14), ya que los datos van cambiando eventualmente tenemos las opciones de (Eliminar/Modificar)

Figura 3. 15 Diagrama listado de comunidades.

Fuente: [Elaboración propia]

En el diagrama (ver Figura3.15), se muestra el formulario de registro de nueva comunidad con los campos debidamente validados.

Figura 3. 16 Diagrama formulario registro comunidades.

Fuente: [Elaboración propia]

En el diagrama (ver Figura 3.16), se muestra el modelo de presentación de creación de un nuevo evento mediante el formulario respectivo.

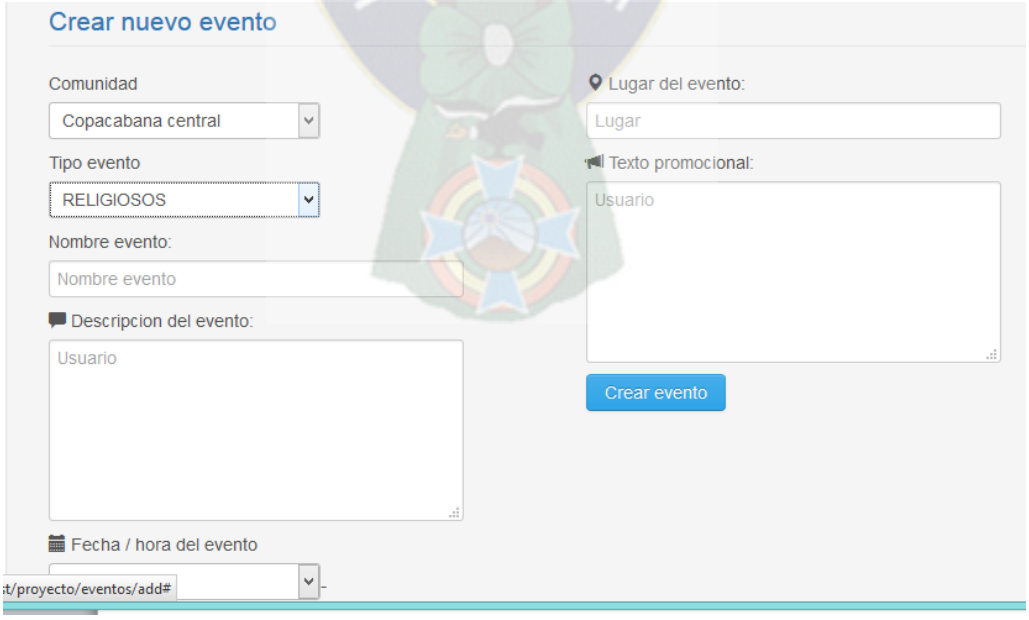


Figura 3. 17 Diagrama formulario registro de evento.

Fuente: [Elaboración propia]

En la figura (ver Figura 3.7), se muestra el modelo de presentación listado de eventos a realizarse.

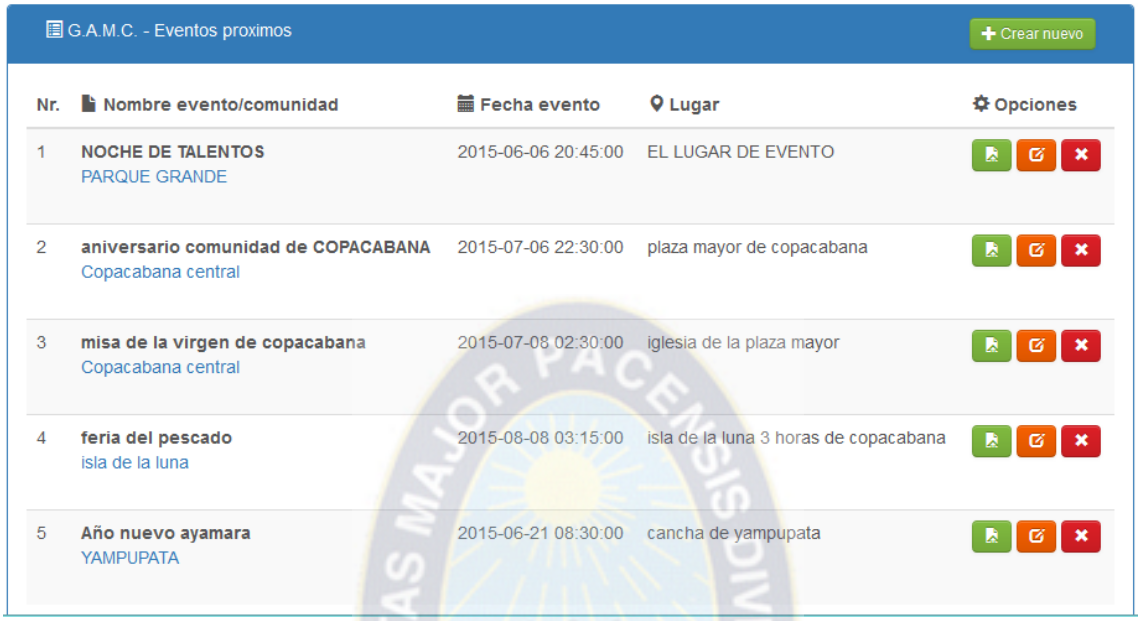


Figura 3. 18 Diagrama listado de eventos a realizarse.

Fuente: [Elaboración propia]

En la figura se muestra el modelo de presentación de registro de auspiciador para un determinado evento.

Figura 3. 19 Diagrama registro de auspiciadores de un evento.

Fuente: [Elaboración propia]

3.3.6 CUARTA ITERACIÓN

A continuación se describen los diagramas de hipertexto y diagramas de presentación para las historias de usuarios que corresponden a la cuarta iteración.

3.3.6.1 Modelo de Hipertexto

En la siguiente diagrama (Ver Figura 3.19) se muestra la estructura del proceso de búsquedas para una ágil obtención de la información almacenada en el sistema.

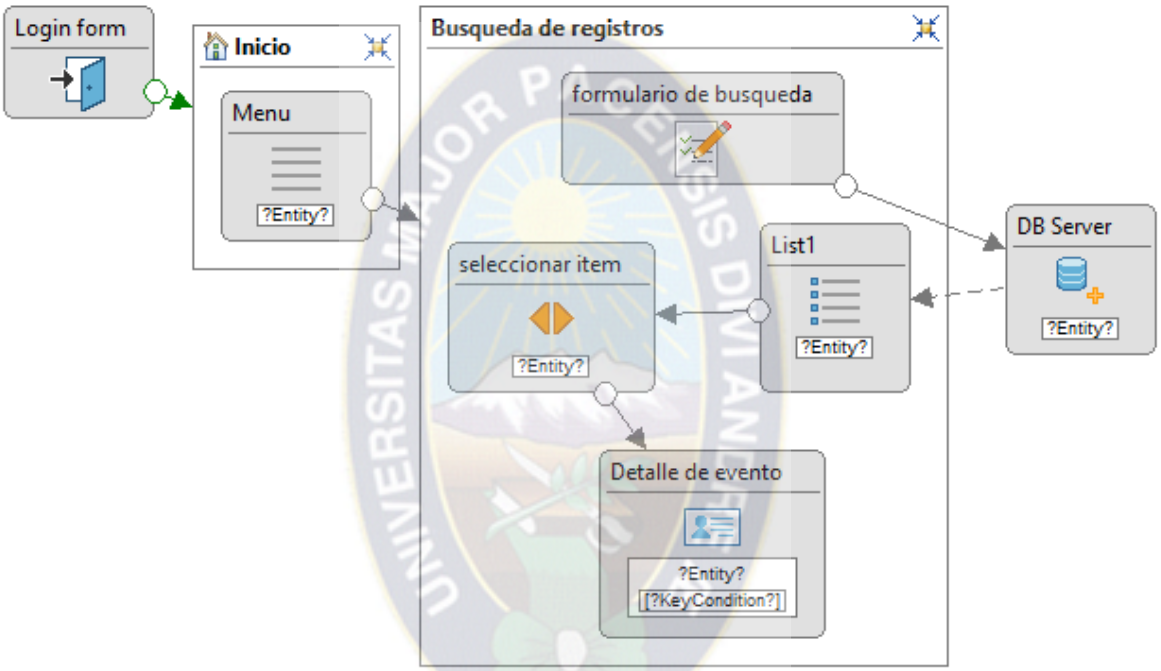


Figura 3. 20 Modelo de hipertexto: Buscador de contenidos.

Fuente: [Elaboración propia]

3.3.6 QUINTA ITERACIÓN

A continuación se describen los diagramas de hipertexto y diagramas de presentación para las historias de usuarios que corresponden a la quinta iteración.

3.3.6.1 Modelo de Hipertexto

En la siguiente diagrama (Ver Figura 3.21) se muestra la estructura del proceso del módulo de reportes y métodos de exportación en documentos digitales.

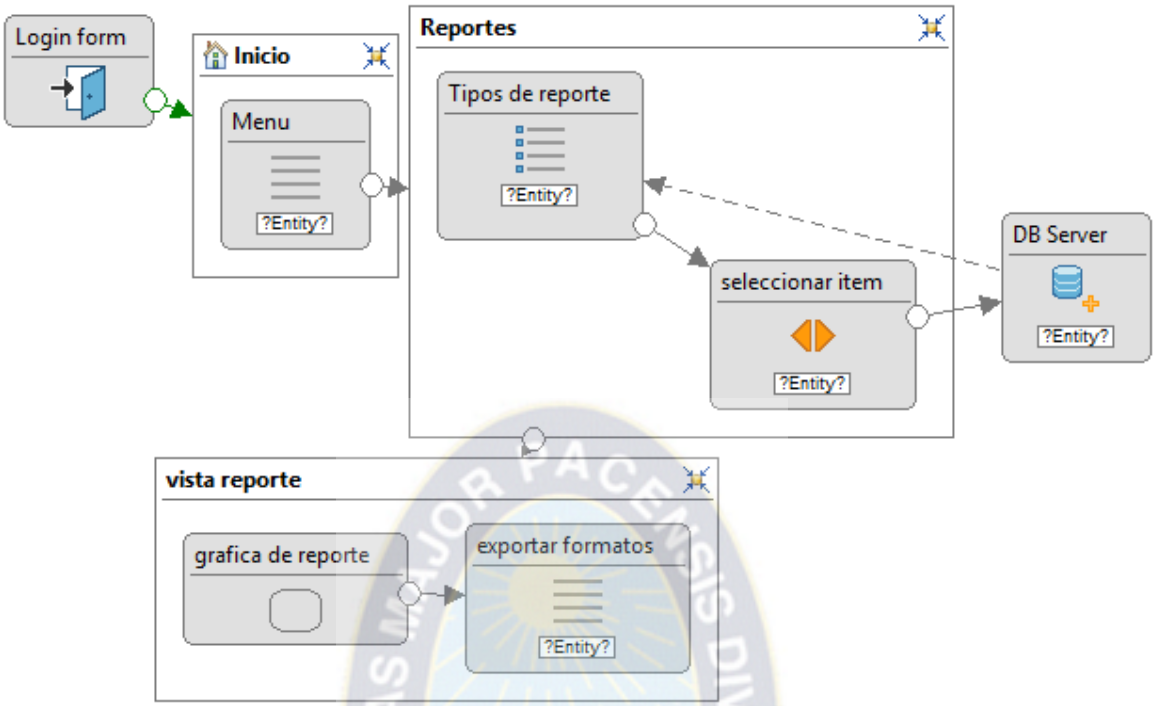


Figura 3. 21 Modelo de hipertexto: Indicadores estadísticas (Reportes).

Fuente: [Elaboración propia]

3.4 DESARROLLO

En esta fase se realizara la programación del Sistema Web acorde al plan de entregas realizada en la fase de planificación, teniendo en cuenta las características que se presentaron y diseñaron

3.4.1 DISEÑO SIMPLE

XP recomienda tener un diseño simple, sencillo y practico al momento de realizar las pantallas para el sistema, este será uno de los aspectos característicos al momento de realizar cada iteración.

3.4.1 PANTALLAS MUERTAS

En la pantalla (ver figura 3.22) se muestra el formulario de autentificación de usuario para ingreso al Sistema Web.

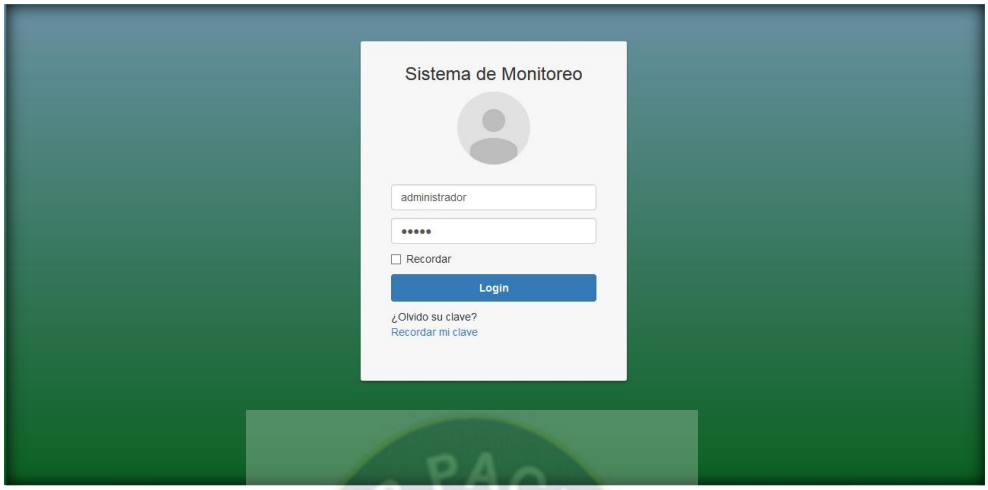


Figura 3. 22 Pantalla de autentificación de usuario

Fuente: [Elaboración propia]

En la pantalla (ver Figura 3.23) se muestra la pantalla principal del sistema con las opciones y privilegios del usuario autentificado.

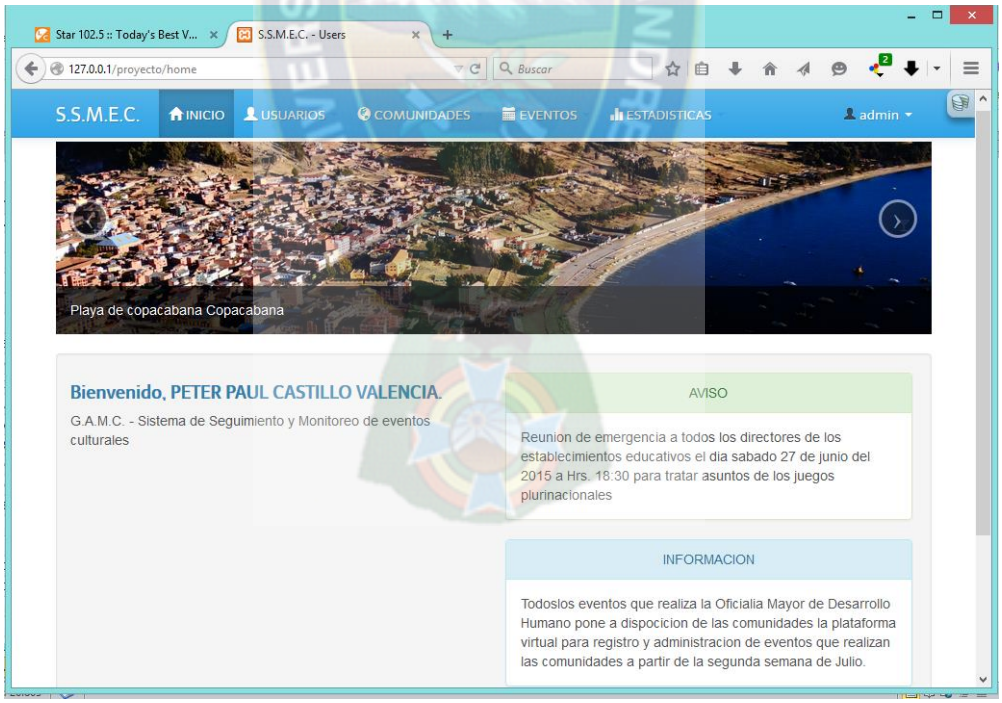


Figura 3. 23 Pantalla principal usuario autentificado

Fuente: [Elaboración propia]

En la pantalla (ver Figura 3.24) se muestra la pantalla en el cual se despliega un formulario para el registro de usuarios por el administrador.

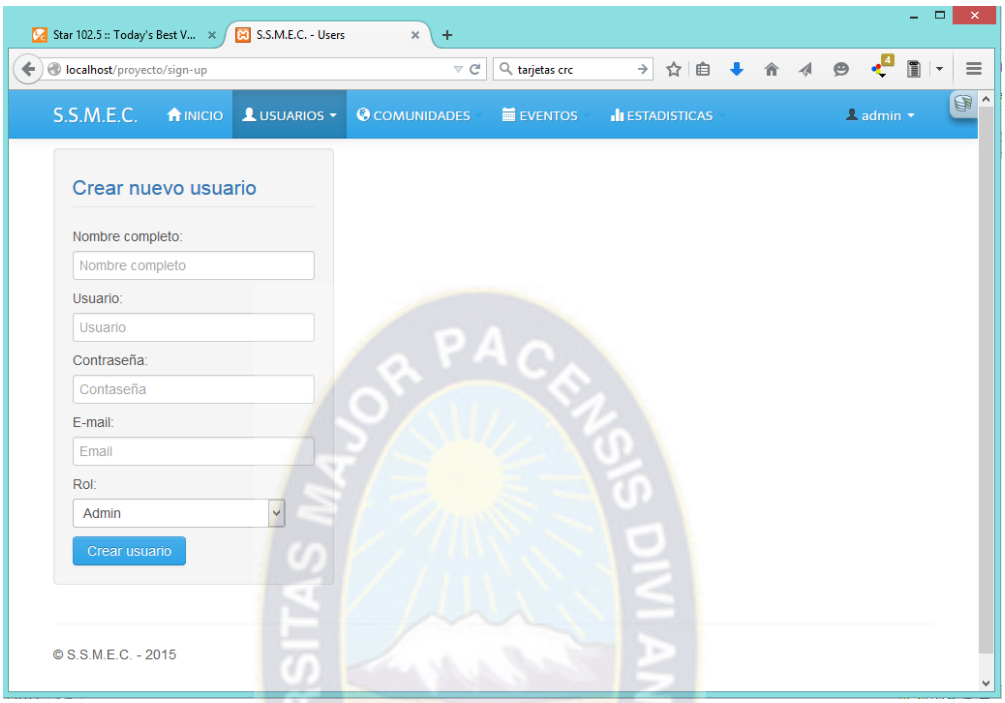


Figura 3. 24 Pantalla formulario de registro usuarios

Fuente: [Elaboración propia]

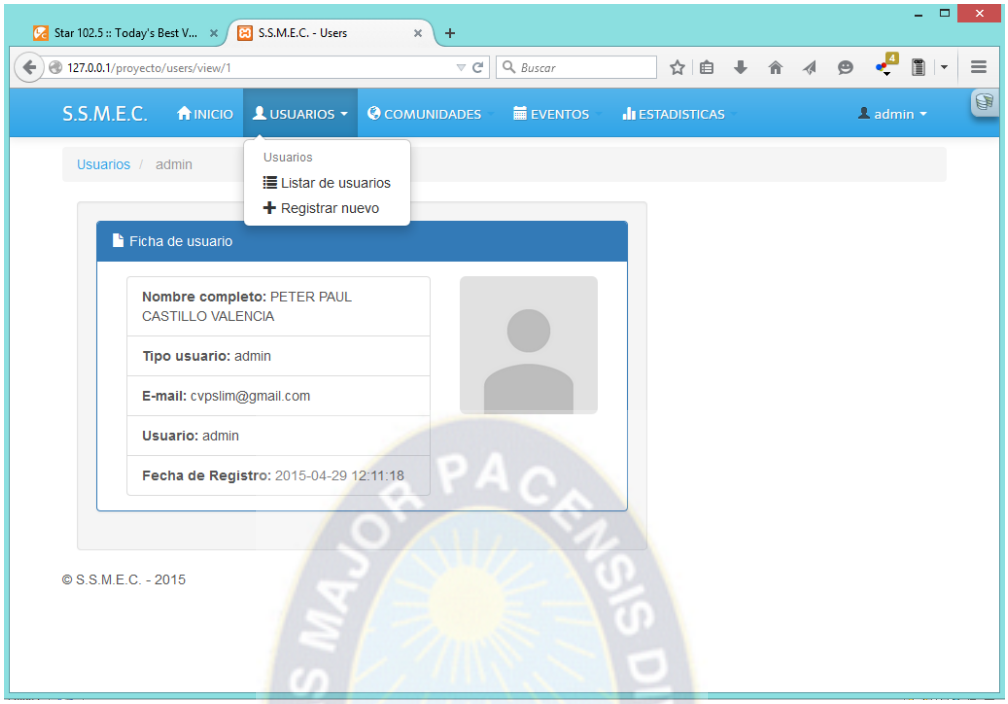


Figura 3. 25 Ficha de registro de usuario

Fuente: [Elaboración propia]

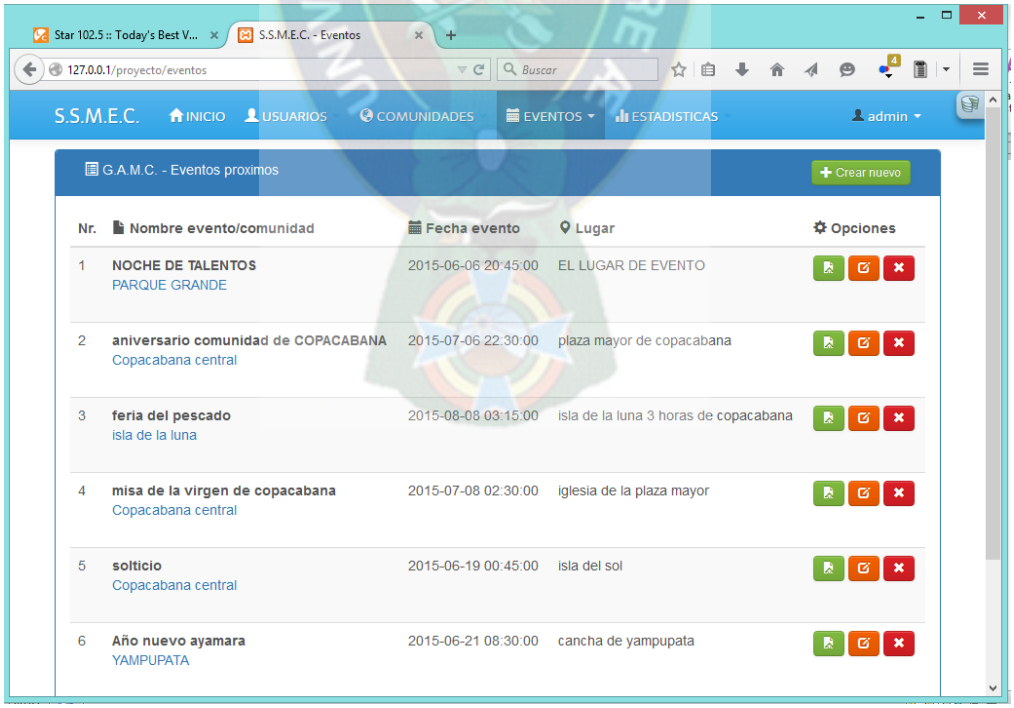


Figura 3. 26 Pantalla listado de eventos a realizarse.

Fuente: [Elaboración propia]

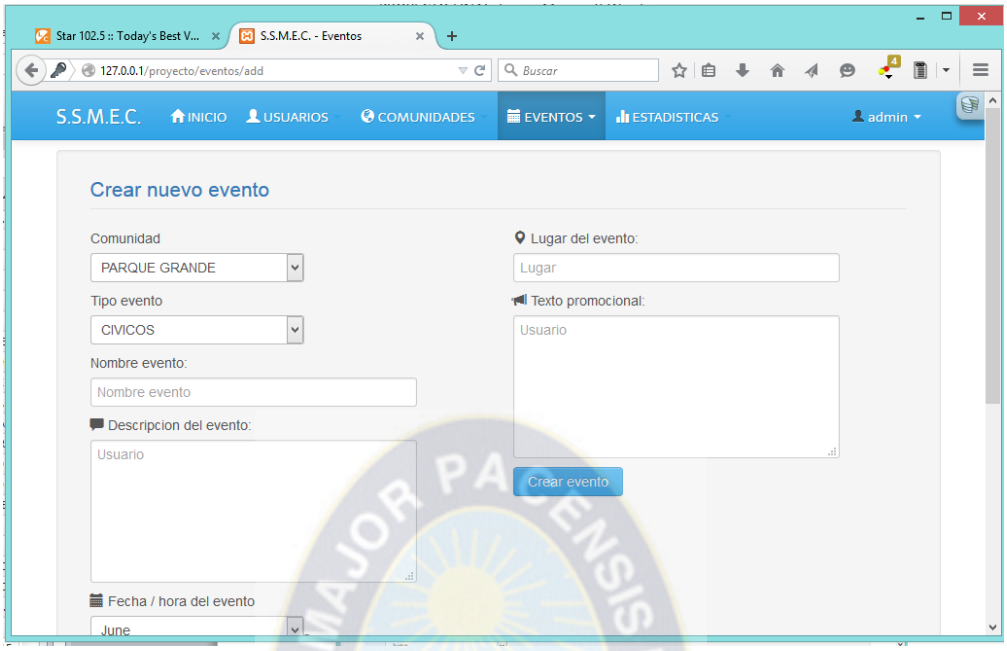


Figura 3. 27 Pantalla de registro de eventos.

Fuente: [Elaboración propia]

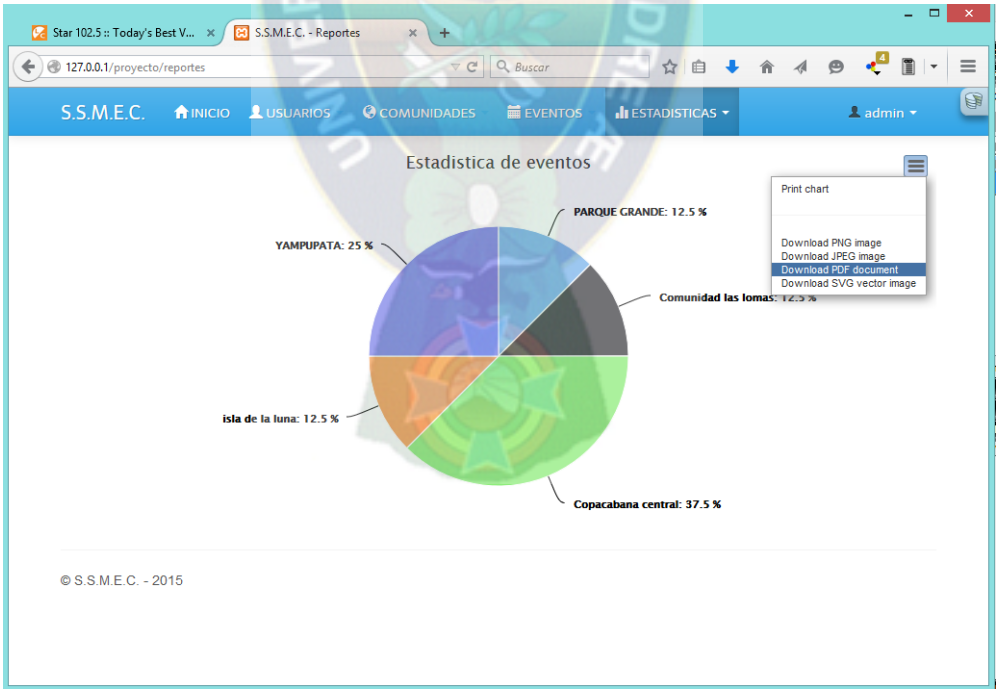


Figura 3. 28 Pantalla módulo de reportes y 4 tipos de exportación

Fuente: [Elaboración propia]

En la pantalla (ver Figura 3.7) se muestra la pantalla y el formulario de registro de nuevos usuarios con los campos correspondientes y debidamente validados en base al tipo de dato.

3.5 PRUEBAS

Para esta fase se van a realizar pruebas a los módulos desarrollados y ya implementados en el sistema, se utilizara las tarjetas de aceptación o también llamadas pruebas de aceptación, a continuación se presentan todas las tarjetas de aceptación, que fueron calificadas por el personal de la Dirección General de Cultura del G.A.M.C.

3.5.1 PRUEBAS DE ACEPTACIÓN

La prueba de Aceptación 1 (ver tabla 3.35) comprueba la ejecución de las funciones de la Historia de Usuario #1 Registro de Usuarios nuevos, que luego de realizar las pruebas correspondientes se pudo obtener el siguiente resultado.

Prueba de Aceptación	
Numero: 1	**Historia de usuario Nro. 1**
Nombre: Registro de Usuarios Nuevos	
Descripción: Realizar el respectivo registro de sus datos en un formulario, en el que se tendrá el control del administrador.	
Condiciones de ejecución: servidor ejecutándose, ingresar al sistema con el usuario y clave correspondiente, ver el menú de opciones y seleccionar la opción de Registro de Usuario nuevo.	
Entrada/Pasos de ejecución: El administrador o usuario encargado se encargar de alimentar el sistema web con la información sobre los usuarios y los permisos que estos deben tener. El sistema actualiza la información en la Base de Datos.	
Resultado esperado: Al ingreso al sistema si el usuario interesado no está registrado, el administrador o usuario autorizado ingresara al sistema y seleccionara la opción de registro de usuario nuevo en el que muestra un formulario para poder llenar sus datos	

respectivos.
Evaluación de prueba: Aceptada

Tabla 3. 35 Verificación de Registro de Usuario

Fuente: [Elaboración propia]

Prueba de Aceptación	
Numero: 2	**Historia de usuario Nro. 3**
Nombre: Registro de eventos culturales	
Descripción: Realizar el registro de un evento cultural a realizarse con sus respectivos datos mediante un formulario con los campos debidamente validados.	
Condiciones de ejecución: servidor ejecutándose, ingresar al sistema con el usuario y clave correspondiente, ver el menú de opciones y seleccionar la opción de eventos y seleccionar crear nuevo evento.	
Entrada/Pasos de ejecución: El administrador o usuario encargado: • ingresa los datos en el formulario y presiona el botón guardar y el evento ya está creado. • Selecciona un evento de la lista y presiona la opción ver detalle y en panel derecho esta las opciones de agregar o eliminar Auspiciadores.	
Resultado esperado: Un usuario registrado tiene la posibilidad de crear un evento y solo tendrá acceso a los eventos que el mismo creo y podrá realizar las actualizaciones de acuerdo al progreso del evento.	
Evaluación de prueba: Aceptada	

Tabla 3. 36 Verificación Modulo eventos

Fuente: [Elaboración propia]

CAPITULO IV

MÉTRICAS DE CALIDAD

En éste capítulo se presenta el proceso realizado para la medición de la calidad del Sistema y los resultados obtenidos.

El objetivo es determinar si el Sistema de Monitoreo para la Dirección General de Cultura del G.A.M.C. puede ser considerado un producto de calidad, aplicando las directivas establecidas en la norma ISO 9126-2.

4.1 ESPECIFICACIÓN DEL MODELO DE CALIDAD

El modelo de calidad que se utilizará para realizar la evaluación es el Modelo de Calidad prescrito en la norma ISO/IEC 9126-2. Las características que componen el modelo son: Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad, Portabilidad.

4.2 SELECCIÓN DE CARACTERÍSTICAS, SUBCARACTERÍSTICAS Y MÉTRICAS A EVALUAR DEL SOFTWARE DE MONITOREO

De acuerdo a las propiedades del Software de Monitoreo y al entorno en el cual va a ser ejecutado, las características y subccaracterísticas de la norma ISO/IEC 9126-2 que fueron seleccionadas con sus respectivos pesos para realizar la evaluación, se encuentran descritos en la Tabla 4.1.

CARACTERISTICA	SUBCARACTERISTICA	PESOS (High/Medium/Low)	OBSERVACION
Funcionalidad	Adecuación	H	
	Exactitud	H	
	Interoperabilidad	N/A	El sistema es independiente, no se relaciona con otros sistemas implantados en la institución.
	Seguridad	M	El sistema está protegido del ingreso no autorizado de personas ajenas a la institución por medio de la combinación de código y contraseña para su

			ingreso. Además, el sistema solo muestra y permite modificar la información a la cual tiene acceso el usuario dependiendo del rol que cumple. Confiabilidad
Confiabilidad	Madurez (hardware/software/datos)	H	
	Tolerancia a fallas	L	Los fallos por corte de energía u otros de este tipo, no afectan en gran medida al funcionamiento ya que el flujo de información no es continuo.
	Recuperabilidad	L	Las copias de seguridad de la base de datos, deben ser realizados manualmente ya que no se cuenta con equipos donde se puedan hacer respaldos automáticamente.
Usabilidad	Entendimiento	M	
	Facilidad de aprendizaje	M	Los usuarios que manejan el sistema son pocos y no existe cambios frecuentes de personal. Además se cuenta con el manual de usuario para el manejo del sistema.
	Operación	M	La operación de los módulos del sistema se manejan en base a perfiles de usuario y el administrador se encarga de realizar las altas, bajas y modificaciones de los objetos entidad del sistema.
	Atractibilidad	M	Los usuarios que manejan el sistema son relativamente pocos y el sistema posee una interfaz que satisface sus actividades.
Eficiencia	Comportamiento en el tiempo	M	El tiempo en realizar las tareas en el sistema no es un factor representativo para el usuario del sistema ya que su uso no es frecuente.
	Utilización de recursos	N/A	No se tomará en cuenta ésta subcaracterística ya que el uso de recursos de entrada y salida son los mínimos para el funcionamiento del sistema.
Mantenibilidad	Capacidad de ser analizado	M	La implementación del sistema está orientado a objetos y se utilizan procedimientos almacenados en la base de datos, por lo se puede identificar fácilmente cualquier error o falla que hubiera en el sistema.
	Facilidad de cambio	L	La estructura del sistema que está orientada a

			objetos, permite adicionar módulos y hacer cambios en los mismos sin afectar internamente al sistema.
	Estabilidad	H	
	Facilidad de pruebas	M	Las correcciones que se realizan en el sistema son pequeñas y no implica tener grandes pruebas a parte de las realizadas para la verificación de cumplimiento de los requerimientos.
Portabilidad	Adaptabilidad	M	El software base para el funcionamiento del sistema (Servidor web Apache, Servidor de Base de Datos MySQL) permite que pueda ser implantado en diferentes plataformas como Windows y Linux principalmente. Por otra parte, ya que el sistema está basado en web, puede ser accedido desde cualquier equipo que cuente con un navegador y acceso a Internet.
	Facilidad de instalación	L	El sistema puede ser instalado en distintas plataformas, pero no es necesario instalarlo en todos los equipos ya que puede ser accedido mediante un navegador web.
	Coexistencia	N/A	El sistema no tiene relación con otro software
	Reemplazabilidad	N/A	En el sistema ha habido mejoras y aumento de funcionalidades, pero no grandes cambios que impliquen crear versiones del sistema

Tabla 4. 1 Características y subcaracteristícas seleccionadas para la evaluación

4.3 OBTENCIÓN DE RESULTADOS DE LAS MÉTRICAS DE CALIDAD EXTERNA DEL SISTEMA

Para realizar el proceso de medición de calidad del Software de Monitoreo, se utilizarán las tabla definidas en la norma ISO/IEC 9126-2.

4.3.1 FUNCIONALIDAD

En la tabla 4.2 se muestra el proceso de medición de la subcaracterística de Adecuación.

Métrica de Adecuación

Subcaracterística	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la	Recursos utilizados

					empresa	
Adecuación	Integración Funcional de la Aplicación	X=1-A/B A=Número de funciones faltantes detectadas en la evaluación B=Número de funciones descritas en los requerimientos de especificación	0<=X<=1 Cuánto más se acerque a 1.0 es lo mejor	A = 0 B = 39 **X = 1**	N/A	Tabla de funciones del sistema (Tabla 3.3)
	Aplicación funcional de cobertura	X=1-A/B A=Número de funciones implementadas incorrectamente o faltantes detectadas en la evaluación B=Número de funciones descritas en los requerimientos de especificación	0<=X<=1 Cuánto más se acerque a 1.0 es lo mejor.	A = 0 B = 39 **X = 1**	N/A	Tabla de funciones del sistema (Tabla 3.3)

Tabla 4. 2 Resultados de las métricas de adecuación

En la tabla 4.3 se muestra el proceso de medición de la subcaracterística de Exactitud.

Métrica de Exactitud

Subcaracterística	**Métricas**	**Fórmula**	**Valor deseado**	**Valores obtenidos**	**Valor estándar establecido en la empresa**	**Recursos utilizados**
Exactitiud	Expectativa de Exactitud	X=A/T A=Número de casos encontrados por los usuarios con una diferencia en contra de los resultados razonablemente esperados T=Tiempo de operación	0<=X Cuánto más se acerque a 0 es lo mejor	A = 0 T = 60 min **X = 0**	N/A	Evaluación con el usuario
	Precisión computacional	X=A/T A=Número de cálculos inexactos que encuentran los usuarios	0<=X Cuánto más se acerque a 0 es lo mejor.	A = 0 T = 60 **X = 0**	N/A	Evaluación con el usuario

		T=Tiempo de operación				
	Precisión	X=A/T A=Número de resultados encontrados por el usuario con un nivel de precisión diferente al requerido T=Tiempo de operación	0<=X Cuánto más se acerque a 0 es lo mejor	A = 0 T = 60 **X = 0**	N/A	Evaluación con el usuario

Tabla 4. 3 Resultados de las métricas de adecuación

4.3.2 CONFIABILIDAD

En la tabla 4.4 se muestra el proceso de medición de la subcaracterística de Madurez.

Métrica de Madurez

Subcaracterí stica	**Métricas**	**Fórmula**	**Valor deseado**	**Valores obtenidos**	**Valor estándar establecido en la empresa**	**Recursos utilizados**
Madurez	Densidad de fallas contra casos de prueba	X=A1/A2 A1=Número de fallas detectadas A2=Número de casos de prueba realizados	0<=X Depende de la fase de pruebas. En las últimas etapas lo menor es mejor	A1 = 5 A2 = 10 **X = 0,5**	N/A	Registro de fallas, errores, caídas del sistema durante un lapso de 4 semanas.
	Resolución de fallas	X=A1/A2 A1=Número de fallas resueltas A2=Número total de fallas detectadas actualmente	0<=X<=1 Cuánto más se acerque a 1.0 es lo mejor cuantas más fallas sean resueltas	A1 = 5 A2 = 5 **X = 1**	N/A	Registro de fallas, errores, caídas del sistema durante un lapso de 4 semanas

Tabla 4. 4 Resultados de las métricas de madurez

4.3.3 USABILIDAD

En la tabla 4.5 se muestra el proceso de medición de la subcaracterística de Entendimiento.

Métrica de Entendimiento

Subcaracterí stica	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados
Entendimiento	Completitud de la descripción	X=A/B A=Número de funciones entendidas B=Número total de funciones	0<=X<=1 El valor más cercano a 1.0 es el mejor	A = 10 B = 11 **X = 0,9**	N/A	Realizar pruebas con el usuario y recopilar los resultados con la información necesaria
	Funciones evidentes	X=A/B A=Número de funciones identificadas por el usuario B=Número total de funciones existentes	0<=X<=1 El valor más cercano a 1.0 es el mejor	A = 11 B = 11 **X = 1**	N/A	Realizar pruebas con el usuario y recopilar los resultados con la información necesaria
	Entendimiento de las funciones	X=A/B A=Número de funciones de interfaces cuyos propósitos son correctamente descritos por el usuario B=Número de funciones disponibles desde la interfaz	0<=X<=1 El valor más cercano a 1.0 es el mejor	A = 10 B = 11 **X = 0,9**	N/A	Realizar pruebas con el usuario y recopilar los resultados con la información necesaria

Tabla 4. 5 Resultados de las métricas de Entendimiento

En la tabla 4.6 se muestra el proceso de medición de la subcaracterística de Aprendizaje.

Métrica de Aprendizaje

Subcaracterí stica	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados
Aprendizaje	Facilidad de aprender las funciones	T=Tiempo medio empleado en aprender a usar una función correctamente	0<T El valor más pequeño es el mejor	**T = 15 min**	N/A	Pruebas con el usuario usando la función de elaboración de cuestionario
	Facilidad de aprender a realizar una tarea	T=Sumar el tiempo de operación hasta que el usuario consiga realizar una tarea específica	0<T El valor más pequeño es el mejor	**T = 20 min**	N/A	Realizar pruebas con el usuario

		dentro de un corto tiempo				

Tabla 4. 6 Resultados de las métricas de Aprendizaje

En la tabla 4.7 se muestra el proceso de medición de la subcaracterística de Operación.

Métrica de Operación

Subcaracterí stica	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados
Operación	Disponibilidad de valores por omisión mientras se usa el sistema	X=1-A/B A=El número de veces que el usuario no logra establecer o seleccionar valores de parámetros en un corto periodo (Porque el usuario no puede usar valores por omisión provistos por el software) B=Número total de veces que el usuario intenta establecer o seleccionar valores de parámetros	0<=X<=1 El valor más cercano a 1.0 es el mejor	A = 0 B = 10 **X = 1**	N/A	Realizar pruebas con el usuario y recopilar los resultados con la información necesaria
	Habilidad de deshacer (Corrección de errores de usuario)	a) X=A/B A=Número de errores de ingreso en los que el usuario corrige exitosamente B=Número de intentos para corregir errores de ingreso b) Y=A/B A=Número de condiciones de error a los que el usuario corrige exitosamente B=Número total de condiciones de error probadas	0<=X<=1 El valor más cercano a 1.0 es el mejor 0<=Y<=1 El valor más cercano a 1.0 es el mejor	A = 11 B = 14 **X = 0,7857** A = 11 B = 14 **X = 0,7857**	N/A	Realizar pruebas con el usuario y recopilar los resultados con la información necesaria

Tabla 4. 7 Resultados de las métricas de Operación

En la tabla 4.8 se muestra el proceso de medición de la subcaracterística de Atractibilidad.

Métrica de Atractibilidad

Subcaracterí stica	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados
Atractibilidad	Interacción atractiva	Cuestionario para valorar el atractivo de la interfaz para el usuario después de tener experiencia en el uso	Depende de los métodos de evaluación del cuestionario	**H**	Cuestionario de evaluación de la interfaz	Realizar un cuestionario con respuestas de opción múltiple sobre la aceptación de la interfaz del sistema

Tabla 4. 8 Resultados de las métricas de Atractibilidad

4.3.4 EFICIENCIA

En la tabla 4.9 se muestra el proceso de medición de la subcaracterística de Comportamiento en el Tiempo.

Métrica de Comportamiento en el Tiempo

Subcaracterí stica	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados
Comportamie nto en el Tiempo	Tiempo de respuesta	T=(Tiempo en el que se obtiene el resultado) – (tiempo en el que se terminó de ingresar el comando)	0 < T El menor valor es el mejor	**T = 0.0167 min**	N/A	Registrar los tiempos de respuesta en las pruebas realizadas al usuario
	Eficiencia para analizar fallos	X=A/T A=Número de tareas completadas B=Periodo de tiempo de observación	0 < X El valor mayor es el mejor	A = 14 B = 60 **X = 0,2333**	N/A	Registrar el resultado de pruebas realizadas al usuario

Tabla 4. 9 Resultados de las métricas de Comportamiento en el Tiempo

4.3.5 MANTENIBILIDAD

En la tabla 4.10 se muestra el proceso de medición de la subcaracterística de Analizabilidad.

Métrica de Analizabilidad

Subcaracterística	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados
Analizabilidad	Capacidad de análisis de errores	X=1-A/B A=Número de fallos donde las causas no han sido encontradas B=Total de errores registrados	0<=X<=1 Mientras más cercano a 1 es mejor	A = 0 B = 5 **X = 1**	N/A	Registro de errores, fallas, caídas del sistema durante un lapso de tiempo de 4 semanas
	Eficiencia para analizar fallos	X=Sum(T)/N	0 <= X	Sum(T)=370 min N = 5 **X = 74**	N/A	Registro de errores, fallas, caídas del sistema durante un lapso de tiempo de 4 semanas

Tabla 4. 10 Resultados de las métricas de Analizabilidad

En la tabla 4.11 se muestra el proceso de medición de la subcaracterística de Estabilidad.

Métrica de Estabilidad

Subcaracterística	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados

Estabilidad	Proporción de cambios exitosos	X=Na/Ta Y=(Na/Ta)/(Nb/Tb) Na=Número de casos en que el usuario encuentra fallas durante la operación después de que el software ha cambiado Nb=Número de casos en que el usuario encuentra fallas antes de que el software sea cambiado Ta=Tiempo de operación durante el periodo de observación especificado después de que el software ha sido cambiado Tb=Tiempo de operación durante el periodo de observación especificado antes de que el software ha sido cambiado	0<X,Y El valor más pequeño y cercano a cero es el mejor	Na = 2 Ta = 29 mins Nb = 1 Tb = 0.5 mins **X = 0.069 Y = 0.034**	N/A	Registro de errores, fallas, caídas del sistema durante un lapso de tiempo de 4 semanas
	Localización del impacto de la modificación	X=A/N A=Número de fallas que aparecen después de que una falla es resuelta por un cambio durante un periodo específico de tiempo N=Número de fallas resueltas	0 <X El valor más pequeño y cercano a cero es el mejor	A = 2 N = 5 **X = 0.4**	N/A	Registro de errores, fallas, caídas del sistema durante un lapso de tiempo de 4 semanas

Tabla 4. 11 Resultados de las métricas de Estabilidad

En la tabla 4.12 se muestra el proceso de medición de la subcaracterística de Facilidad de prueba.

Métrica de Facilidad de prueba

Subcaracterística	Métricas	Fórmula	Valor deseado	Valores obtenidos	Valor estándar establecido en la empresa	Recursos utilizados

Facilidad de prueba	Eficiencia de la repetición de pruebas (re-test)	X=SUM(T)/N T=Tiempo usado en probar para asegurar que los reportes de fallos fueron resueltos o no. N=Número de fallas resueltas	0<X El valor más pequeño es mejor	Sum(T) = 75 mins N = 5 X = 15 mins	N/A	Registro de errores, fallas, caídas del sistema durante un lapso de tiempo de 4 semanas

Tabla 4. 12 Resultados de las métricas de Facilidad de prueba

4.3.6 PORTABILIDAD

En la tabla 4.13 se muestra el proceso de medición de la subcaracterística de Adaptabilidad.

Métrica de Adaptabilidad

Subcaracterística	**Métricas**	**Fórmula**	**Valor deseado**	**Valores obtenidos**	**Valor estándar establecido en la empresa**	**Recursos utilizados**
Adaptabilidad	Adaptabilidad en ambiente de Hardware	X=1-A/B A=Número de funciones operacionales que sus tareas no son completadas o no cumplen con adecuados niveles durante la prueba de operaciones de hardware B=Total de funciones probadas	0<=X<=1 El valor más grande es mejor	A = 0 B = 7 **X = 1**	N/A	Realizar pruebas de operación de la funcionalidad de ciertas funciones
	Integración funcional de la aplicación	T=Suma del tiempo utilizado por el usuario que opera el Software para completar la adaptación de Software con el ambiente del usuario, cuando el usuario intenta instalar o cambiar configuraciones	0 < T Mientras el valor es pequeño es mejor	**T = 0.05**	N/A	Tomar el tiempo utilizado por el usuario para instalar o cambiar configuraciones en el sistema

Tabla 4. 13 Resultados de las métricas de Adaptabilidad

4.4 DETERMINACIÓN DE LA CALIDAD DEL SISTEMA

En la Figura 4.1 se muestra la tabulación de todos los datos obtenidos durante el proceso de evaluación del Sistema de Monitoreo de eventos culturales basado en el modelo de calidad externa de la norma ISO 9126.

En el gráfico se puede observar que las subcaracterísticas ´

Poseen valores aceptables de calidad.

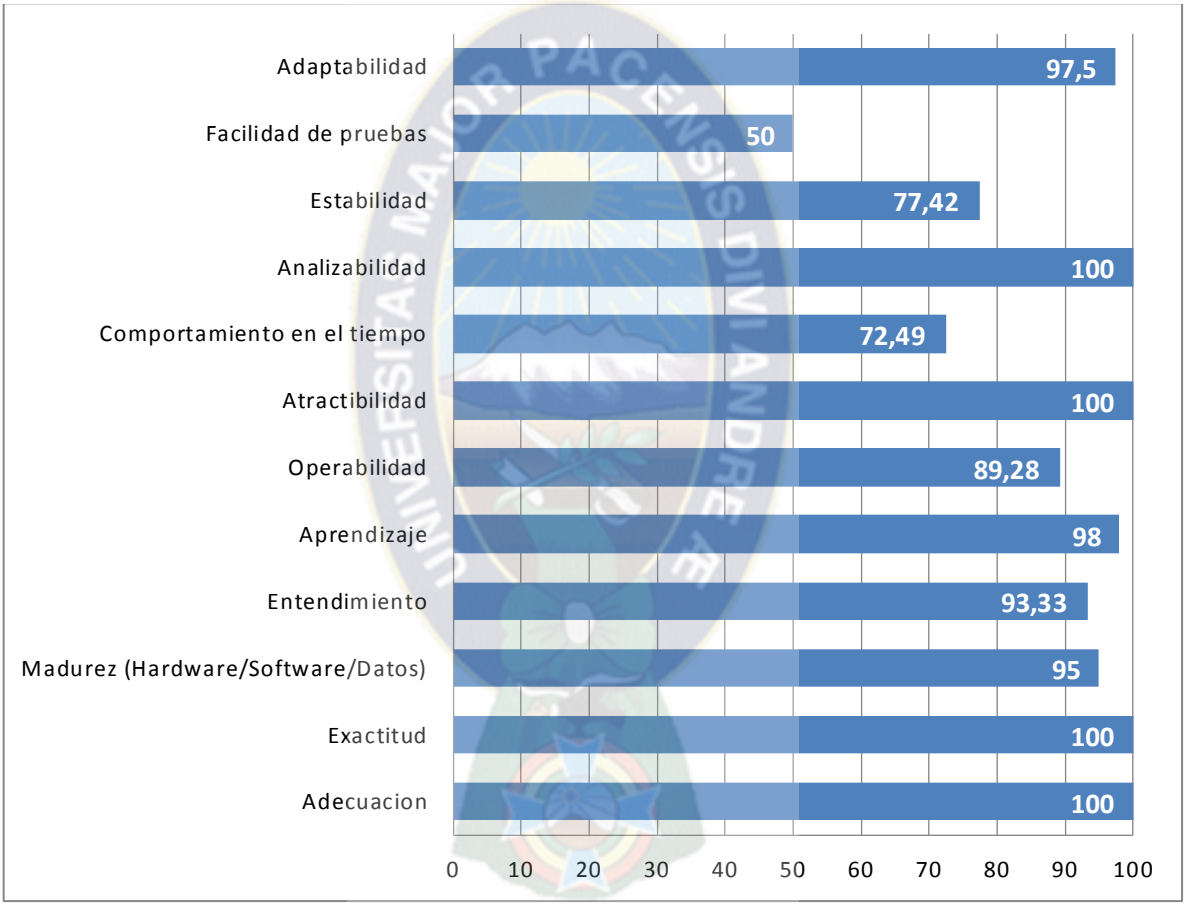


Figura 4. 1 Tabulación de resultados de la medición de calidad

la Figura 4.2 se muestran los resultados del nivel de calidad obtenido por el Sistema para el Monitoreo de eventos culturales del G.A.M.C., en donde se puede ver que cumple con el 89.42 % de las características de calidad seleccionadas, encontrándose dentro del nivel de aceptabilidad, por lo tanto satisface los requisitos de calidad.

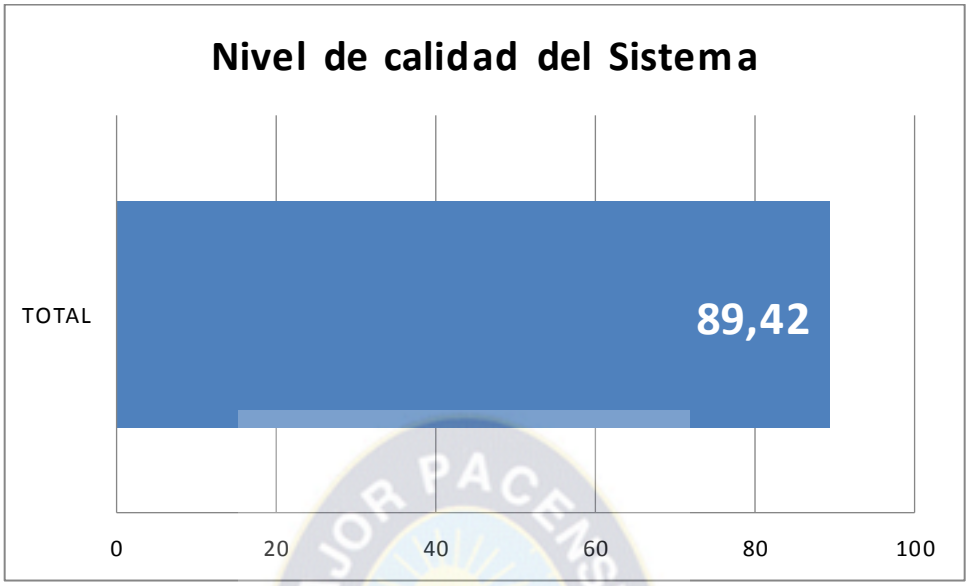


Figura 4. 2 Resultado del nivel de calidad del Sistema de Seguimiento y Monitoreo.

4.5 INTERPRETACIÓN DE LOS RESULTADOS

De los resultados obtenidos en el proceso de medición de la calidad del Sistema para el Monitoreo de Eventos Culturales, se obtiene que el nivel de Funcionalidad es el más Alto, como se puede apreciar en la Figura 4.3, por tanto cumple las expectativas del personal responsable del sistema.

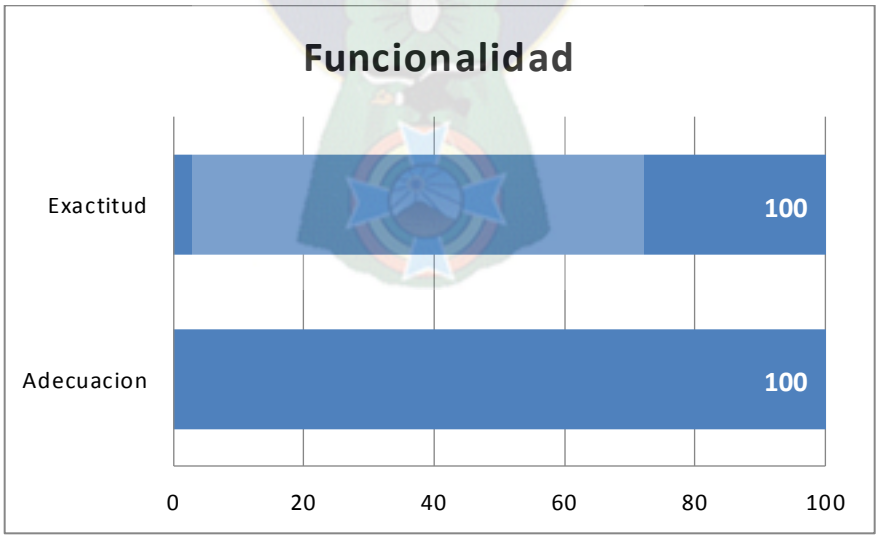


Figura 4. 3 Resultados del nivel de calidad de Funcionalidad

El nivel de confiabilidad del Sistema es Alto, como se puede apreciar en la Figura 4.4. Se puede observar que para ésta característica solo se tomaron en cuenta las métricas de la subcaracterística de Madurez ya que para el entorno en el cual se implementó el sistema, las subccaracterísticas de Tolerancia a fallas y Recuperabilidad no tienen mucha relevancia.

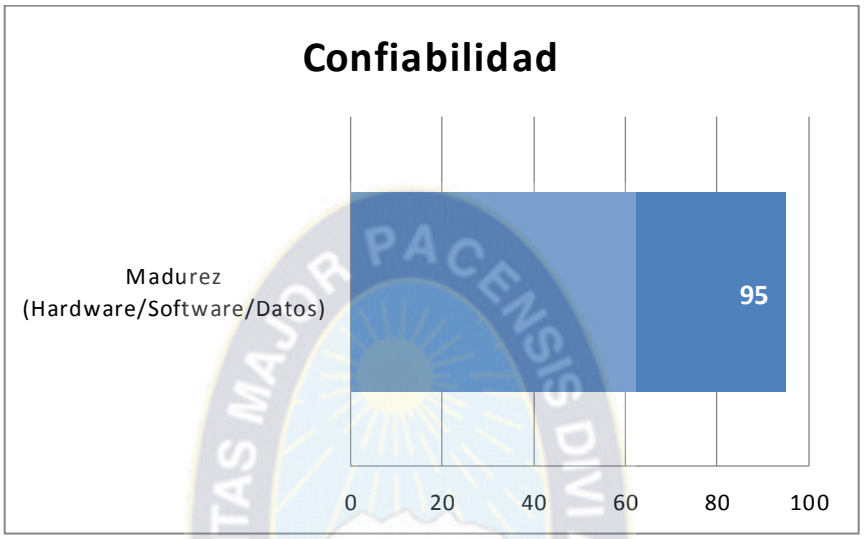


Figura 4. 4 Resultados del nivel de calidad de Confiabilidad

El nivel de Usabilidad es alto, como se puede apreciar en la Figura 4.5, por tanto cumple con las expectativas del personal encargado de monitoreo y puede ser fácilmente comprendido y utilizado por el personal de la institución.

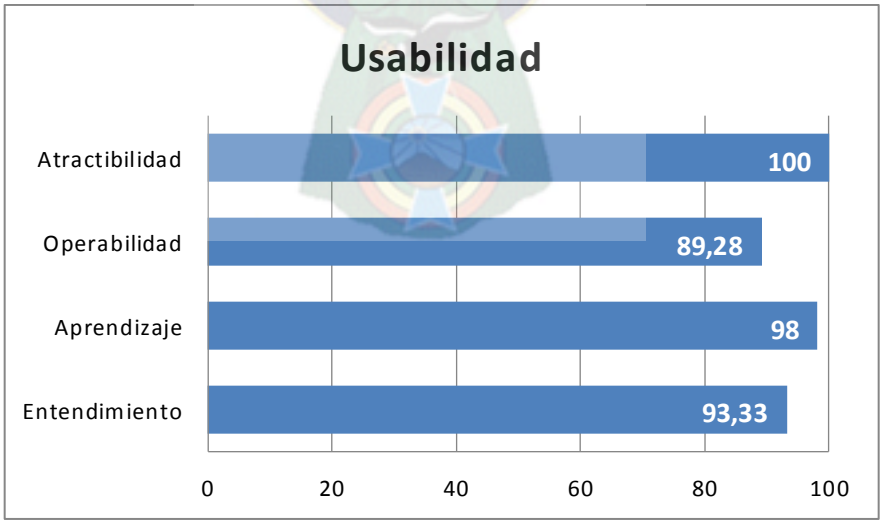


Figura 4. 5 Resultados del nivel de calidad de Usabilidad

El nivel de Eficiencia es alto, como se puede apreciar en la Figura 4.6, por tanto cumple con el nivel requerido por el personal responsable del sistema, a pesar de que se podría mejorar éste nivel obtenido estableciendo un procedimiento para analizar los fallos obtenidos y manejando una base de conocimientos sobre los problemas reportados hasta el momento.

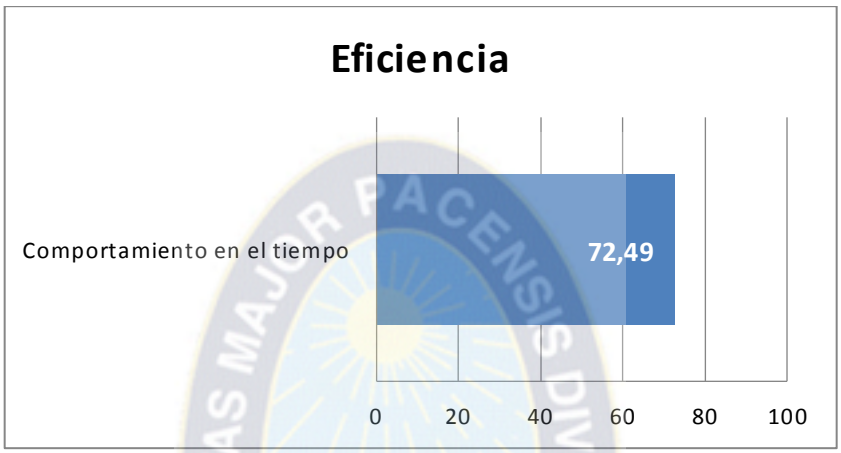


Figura 4. 6 Resultados del nivel de calidad de Eficiencia

El nivel de Mantenibilidad es alto, como se puede apreciar en la Figura 4.7, sin embargo el valor de "Facilidad de pruebas" obtuvo un valor medio en el proceso de medición, es por esto que se debe implementar mejores procesos y procedimientos para las pruebas del sistema.

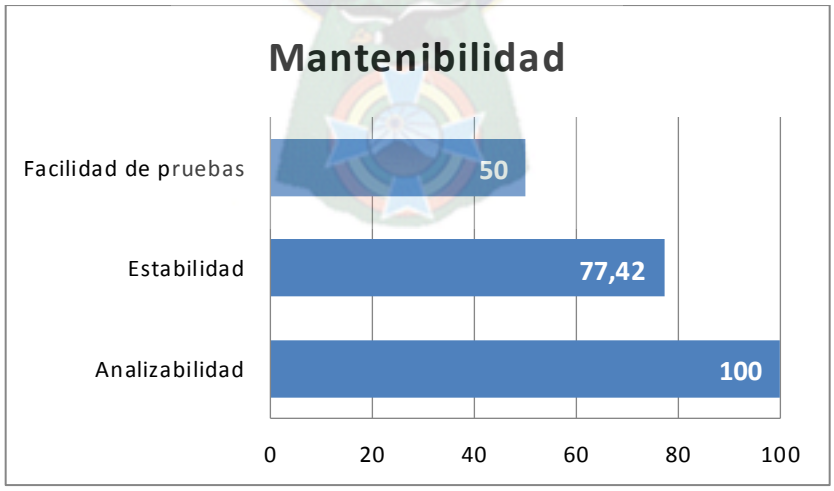


Figura 4. 7 Resultados del nivel de calidad de Mantenibilidad

El nivel de Portabilidad es alto, como se puede apreciar en la Figura 4.8, por tanto el sistema puede adaptarse a diferentes plataformas de hardware y software. Puede funcionar bajo diferentes sistemas operativos como Windows y Linux debido a que las herramientas utilizadas para su desarrollo como el servidor web Apache y el Gestor de Base de Datos MySQL están disponibles para éstos y otros sistemas operativos.

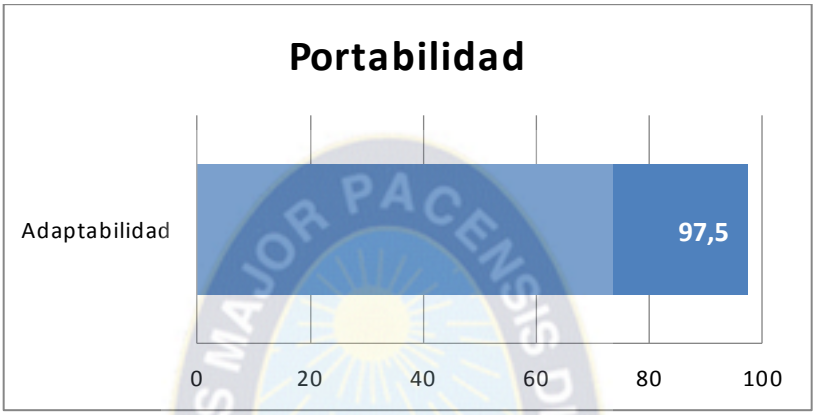


Figura 4. 8 Resultados del nivel de calidad de Portabilidad

CAPITULO V

COSTOS Y BENEFICIOS

En este capítulo se hace un cálculo del costo de desarrollo del Sistema de Información para el Seguimiento y Monitoreo del CCTS haciendo uso del modelo de estimación COCOMO.

5.1. EVALUACION DE COSTOS DE PROGRAMACION

El modelo COCOMO utiliza, para la estimación de un proyecto, el número de instrucciones "fuente" necesarios para su desarrollo. Generalmente se excluyen todas aquellas instrucciones que no pertenezcan estrictamente al código que va a ser ejecutado. Por "instrucciones fuente" se entienden todas aquellas instrucciones desarrolladas por el Equipo de Proyecto, y que deberán ser posteriormente convertidas en código ejecutable por el ordenador. Se excluye por tanto toda clase de comentarios existentes en los programas existentes, por ser siempre constantes.

Para medir el esfuerzo medido en técnicos – mes, el modelo COCOMO básico utiliza la siguiente ecuación:

$$E = a * \mathrm{KLOC}^{b}$$

Donde E es el esfuerzo aplicado para el desarrollo del proyecto, KLOC es el número de miles de líneas de código del proyecto. Los coeficientes a y b y se obtienen de la siguiente tabla:

Tipo de proyecto	***a***	***b***
Orgánico	2.4	1.05
Semiacoplado	3.0	1.12
Empotrado	3.6	1.20

Tabla 5. 1 Valores de los coeficientes a y b del modelo COCOMO por tipo de proyecto

Fuente: Boehm

La duración del proyecto, se calcula a partir del esfuerzo (E):

D = c * E^d

Donde D es la duración del proyecto medido en meses, y los coeficientes c y d se obtienen de la siguiente tabla:

Tipo de proyecto	***c***	***d***
Orgánico	2.5	0.38
Semiacoplado	2.5	0.35
Empotrado	2.5	0.32

Tabla 5. 2 Valores de los coeficientes c y d del modelo COCOMO por tipo de proyecto

Fuente: Boehm

Utilizando la herramienta "LinesOfCodeWichtel" para el conteo de líneas de código fuente, se obtuvieron los siguientes resultados:

Líneas 40472

Líneas de código 25049

Líneas blancas 9565

Líneas comentadas 5858

Estos resultados no toman en cuenta el código de los procedimiento almacenados en la base de datos. El conteo de líneas de código de procedimientos almacenados es el siguiente:

Líneas de código 2050

Sumando las líneas de código ejecutable PHP y las líneas de código de los procedimientos almacenados, se tiene el siguiente resultado:

Total líneas de código 27099

Aplicando el modelo COCOMO básico al proyecto, y usando un tipo de proyecto orgánico obtenemos una estimación para el esfuerzo:

$E = 2.4 * KLOC^{1.05}$
$E = 2.4 * (27.1)^{1.05}$
E = 77 técnicos-mes

Para calcular la duración del proyecto usamos la estimación de esfuerzo:

D = 2.5 * E0.38

D = 2.5 * 770.38

D = 13 meses

El valor de la duración del proyecto permite recomendar un número de técnicos N para el proyecto. N = E / D

N = 77 / 13

N = 6 técnicos

El salario de un programador puede oscilar entre sus 200 Dólares Americanos, cifra que será tomada en cuenta para la estimación siguiente:

*Costo del Software Desarrollado = Número de Programadores * Salario de un Programador*

Reemplazando los datos tenemos:

Costo del software desarrollado = 6 * 200

Costo del software desarrollado = 1200 Dólares Americanos.

5.2 COSTO DE ELABORACION DEL PROYECTO

Además de los costos de programación, para la elaboración del proyecto se debe tomar en cuenta los costos en las etapas de requerimientos, análisis, diseño y pruebas. El tiempo empleado para estas actividades es el siguiente:

Planificación 2 meses Diseño 3 meses Pruebas 2 semanas Considerando el sueldo para el analista de 300 dólares (este valor es muy subjetivo, ya que el sueldo depende mucho de su experiencia), el costo de éstas actividades, será de 1650 dólares.

La tabla 5.3, resume los costos para la elaboración del proyecto:

DESCRIPCION	COSTO MENSUAL	MESES	SUBTOTAL
Planificación	300	2	600 $us
Diseño	300	3	900 $us
Desarrollo	200	6	1200 $us
Pruebas	300	0.5	150 $us
TOTAL		**2850 $us**	

Tabla 5. 3 Tabla de costos del Sistema

El costo estimado para la elaboración del Proyecto es de 2850 dólares.

5.3 BENEFICIOS

La implantación del Software de Información para el seguimiento y monitoreo de eventos culturales de la *Dirección General de Cultura del G.A.M.C.*, conlleva los siguientes beneficios:

- ✓ Se agiliza la recolección de información a través de cuestionarios electrónicos elaborados por los encargados de monitoreo, que pueden ser llenados por los administrativos y personas encargadas de los eventos desde cualquier equipo de computación dentro la institución. El tiempo empleado para el llenado y supervisión de cada evento antes de la implementación del sistema era de aproximadamente 5 horas. y 20 min., con el uso del sistema, éste tiempo se redujo a 20 min. solo para el registro de los eventos ya que la disponibilidad de información se realiza de inmediato.
- ✓ Se minimiza los costos de materiales utilizados para la elaboración y llenado de eventos. Los materiales utilizados para la elaboración de cada evento (papel, fotocopias, impresiones, etc.), requerían un monto de Bs. 68.80 aproximadamente, con la implementación del sistema ya no es necesario utilizar ninguno de estos materiales, excepto el de las impresiones para los informes.
- ✓ Toda la información concerniente a la ejecución de programas y eventos culturales de todas las comunidades pertenecientes al municipio de Copacabana está centralizada en una sola base de datos.
- ✓ Con los reportes que genera el sistema, se consigue una retroalimentación oportuna acerca de eventos realizados para nuevas proyecciones, además del impacto que tienen en la sociedad.

CAPITULO VI

SEGURIDAD

En este capítulo definiremos los procedimientos de seguridad y operación necesarios para garantizar el funcionamiento correcto del software y poder garantizar el cumplimiento de los niveles de servicios que exigirá el software en cuanto a la gestión de operaciones. Tomando como referencia los requisitos establecidos para el sistema, y teniendo en cuenta la arquitectura y las características del entorno tecnológico, se lleva a cabo la definición de los requisitos de seguridad y control de acceso necesarios para garantizar la protección del sistema y minimizar el riesgo de pérdida, alteración o consulta indebida de la información. Para ello, se diseñan los procedimientos relacionados con:

- Acceso al sistema y sus recursos.
- Mantenimiento de la integridad y confidencialidad de datos.
- Copias de seguridad y recuperación de datos.
- Recuperación y reanudación de trabajos.

6.1. ACCESO AL SISTEMA Y A SUS RECURSOS

El sistema cuenta con una base de datos relacional para almacenar sus datos y ejecutar determinadas transacciones. El acceso a dichos datos, estructura de datos y transacciones se encuentra protegido por el mecanismo de autenticación básica provisto por el gestor de base de datos. En este caso la base de datos provee un sistema de seguridad basado en usuario y contraseña y un mecanismo que permite legislar los equipos desde los cuales es posible conectarse, o bien, delegar la autenticación al sistema operativo. La administración de la base de datos está a cargo de una sola persona que es el encargado de sitemas del G.A.M.C. Los demás usuarios solo pueden acceder a los datos a través del sistema con la combinación de su código de usuario y contraseña.

Todos los actores que intervienen en el diagrama de validación de usuario del sistema, son usuarios que podrán acceder al sistema con distintos privilegios. Cada uno de los actores,

representan los roles bajo los cuales están agrupados los usuarios para acceder al sistema.

En la Figura 6.1 se presenta el diagrama de validación de usuario. El sistema permite el acceso únicamente a los usuarios registrados que cuentan con su código y contraseña.

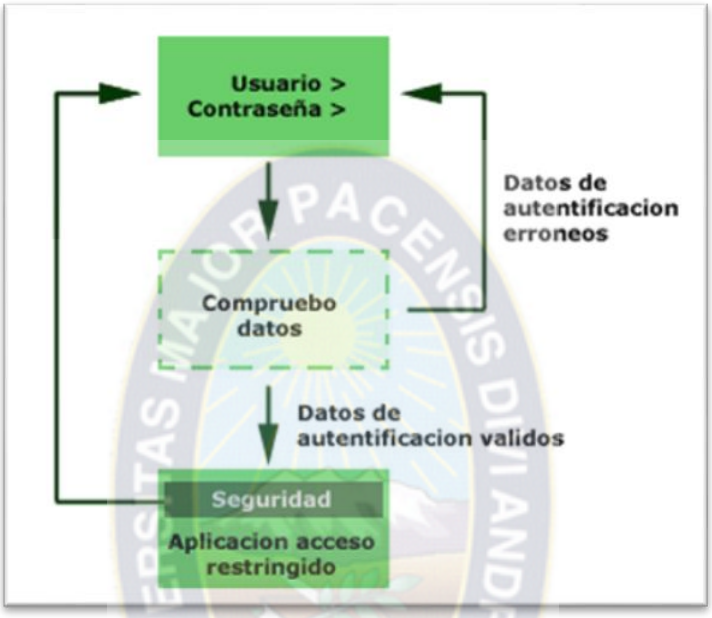


Figura 6. 1 diagrama validación de usuario

Fuente: Interpretación propia

6.2. MANTENIMIENTO DE LA INTEGRIDAD Y CONFIDENCIALIDAD DE DATOS

La confidencialidad de los datos se obtiene debido a que el sistema no permite ver los datos que no estén relacionados con el usuario autenticado que está usando el sistema. Cuando un usuario ingresa al sistema con su código y contraseña, el sistema carga el menú de opciones a los cuales puede acceder.

Por otro lado, la base de datos se encuentra protegida dentro del servidor de aplicaciones en una carpeta con acceso restringido.

6.3. COPIA DE SEGURIDAD Y RECUPERACIÓN DE DATOS

Las bases de datos relacionales provén mecanismos específicos para los resguardos de seguridad y la recuperación ante una eventual necesidad. Estos mecanismos varían desde el backup a nivel de sistema de archivo hasta copias replicadas en línea para cambiar el servidor de base de datos y continuar operando sin interrupciones. Dado que este proyecto está desarrollado para la Dirección General de Cultura del G.A.M.C. dentro del marco de un trabajo institucional los presupuestos deben ser aprobados por el concejo municipal, es por tanto que la dirección no cuenta con presupuesto alguno, por tanto se recomienda un esquema de backup semanal basado en copias de la base de datos y del sistema de archivos en CD-ROM.

6.4. RECUPERACIÓN Y REANUDACIÓN DE TRABAJOS

Los trabajos no pueden ser recuperados o reanudados. Simplemente, cada operación puede ejecutarse satisfactoriamente, o no. En caso de una interrupción en el servicio sucederá lo indicado en la Tabla 6.1.

Origen de la interrupción	Consecuencias
Fallo en el cliente que ocasiona la caída del componente mismo.	El sistema no guarda estado de sus objetos en el cliente. Con lo cual, cualquier operación no terminada (enviada al servidor) será deshecha por completo. Al reanudar el uso del sistema, el usuario deberá volver a ejecutar todos los pasos hechos anteriormente.
Fallo en el componente del servidor, que ocasiona la caída del mismo.	El sistema tampoco almacena estado en el componente servidor. En caso de caerse éste componente, se producirá la caída de todo el sistema.
Fallo en el componente base de datos, que ocasiona la caída del mismo.	La base de datos posee lógica de gestión de transacciones. Por tanto, una caída en éste componente resultará en volver las transacciones abiertas al último estado consistente anterior.

Tabla 6. 1 Consecuencias de interrupciones en el software

Fuente: Elaboración propia

CAPITULO VII

CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

Después de haber culminado con el desarrollo del proyecto, y abiendo cumplido con los objetivos planteados en el Capítulo I, se llega a las siguientes conclusiones:

- Se desarrolló y se implementó el Software de Seguimiento y Monitoreo de eventos culturales de la Dirección General de Cultura del G.A.M.C. siguiendo todas las fases de la Metodología XP, llegando a cumplir con todos los requisitos y expectativas de la Institución.

- El sistema está desarrollado para ejecutarse en un entorno web, por lo cual puede ser accedido desde cualquier equipo que cuente con acceso a Internet.

- El tiempo empleado para el proceso de recolección de información a través de cuestionarios electrónicos, se redujo en un 93.75% con el uso sistema.

- Con la implementación del sistema, el costo de los materiales utilizados para la planificación de eventos (papel, impresión, fotocopias, etc.), se redujo en un 98.5%.

- El sistema permite un análisis y retroalimentación oportuna a través de reportes disponibles en cualquier momento.

- El sistema cuenta con claves de acceso de seguridad para los usuarios registrados, donde la contraseña es encriptado por el algoritmo MD5. Los usuarios pueden acceder únicamente a la información que le corresponde, de esta manera se mantiene la seguridad de la información.

- La implementación del módulo de administración, permite realizar las altas, bajas, modificaciones y listado de Comunidades, Eventos, Auspiciadores, Usuarios.
- La utilización de la metodología de desarrollo XP, permitió cumplir satisfactoriamente con los requisitos planteados y no hubo muchas variaciones respecto al tiempo estimado para el desarrollo del proyecto.
- Se logró centralizar la información de los eventos que realiza la Direccion General de Cultura del G.A.M.C. través de una base de datos bien estructurada.
- Siendo el Software de Seguimiento y Monitoreo de Eventos Culturales, una aplicación web con interfaz amigable, de fácil manejo y desarrollado por módulos, el tiempo de capacitación al usuario es relativamente corto.

7.2 RECOMENDACIONES

Se recomienda para que los usuarios finales logren obtener un mayor beneficio por parte del sistema tomar en cuenta los siguientes puntos:

- Mantener actualizada la información ingresada en el sistema respecto a eventos, usuarios y comunidades pertenecientes al municipio de Copacabana para que el seguimiento a los eventos que se realizan sea confiable.
- Contar con respaldo físico de las notas ingresadas en el sistema por los usuarios que promueven los eventos.
- Se recomienda para implementar nuevos módulos en el sistema mantener la estructura MVC para que el mantenimiento no conlleve a gastos innecesarios en un futuro.
- Verificar constantemente el registro de usuarios que no pertenecen a la institución por razones de seguridad.
- Realizar copias de seguridad de la base de datos cada cierto tiempo por razones de seguridad de información.
- Proveer el manual de a los futuros usuarios del sistema para que manipulen correctamente y se produzca un excelente rendimiento de la aplicación.

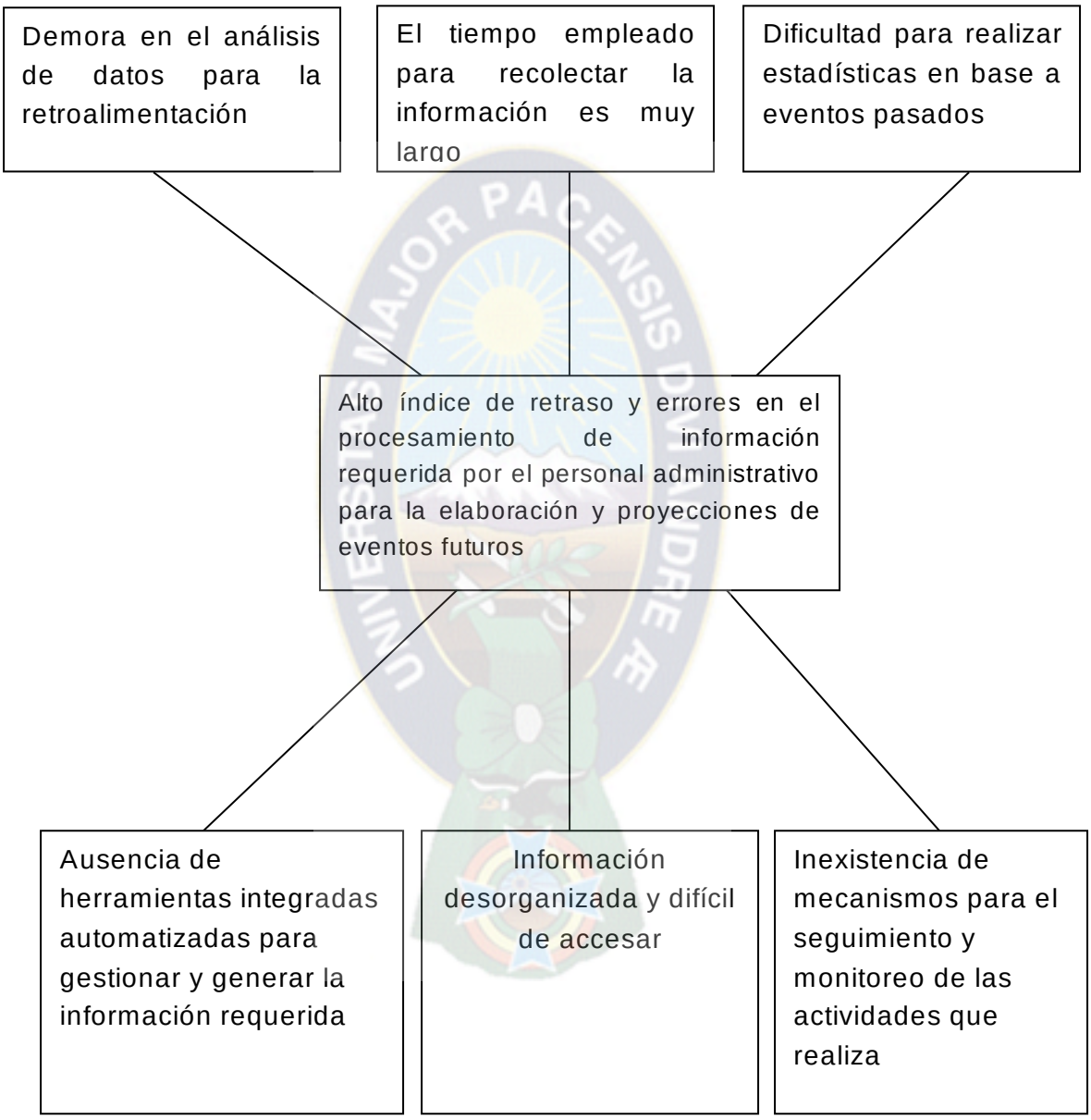
BIBLIOGRAFIA

- ✓ [Press, 1993] PRESSMAN ROGER, 1993, “ingeniería de Software”, Edit McGraw Hill Interamericana, 3ra edición, España 1993.
- ✓ [Larm, 1999] Larman Graing, 1999, UML y Patrones, Introducción al análisis dediseño orientada a objetos, Pags. 506, 1ª Edición, Prentice_Hall Hispanoamereicana S.A. Mexico.
- ✓ [KEND, 1997] Kendall & Kendall. Análisis y Diseño de Sistemas, Págs. 913, 3ªEdición, Prentice_ Hall Hispanoamericana S.A. México.
- ✓ [You, 1993], YOURDON EDWARD, 1993, “Análisis Estructurado Moderno”, Prentice Hall Hispanoamericana S.A., Respecto a la primera edición, México 1993.
- ✓ [CORR, 2005] CORRALES, Leana. Seguimiento, Monitoreo y Evaluación de Proyectos Sociales. CURN / CICAP, 2005.
- ✓ [WEL, 2008] WELT HUNGER HILFE, Orientacion a efectos e impactos. Parte I: Transfondo y definiciones. Deutsche Welthungerhilfe e. V., 2008.
- ✓ [JAC, 2000] JACOBSON, Ivar. BOOCH, Grady. RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Madrid, España: Addinson-Wesley Iberoamericana España S.A. 2000.
- ✓ [SCHACH, 2005] Stephen R. Análisis y diseño orientado a objetos con UML y el proceso unificado. tr. Lorena Peralta Rosales. México:McGraw-Hill, 2005.
- ✓ [BOEHM, 1981] Barry W. Boehm. Software Engineering Economics, Pretince Hall, Englewood Cliffs, NJ.
- ✓ [Gonzáles 05] Desarrollo Web con PHP y MySQL Ing. Joel Gonzáles Estrada
- ✓ [Press,1998] Roger S. Pressman 2003, Ingeniería del Software un Enfoque práctico, Quinta Edición
- ✓ [Plati,2003] Platitini Velthuis M. & Garcia Rubi o F.J., Calidad en el Desarrollo y mantenimiento del Software, 2003
- ✓ [Rosen,1998] “Ciencia para todos”, Colegio Nacional y fondo de cultura, México
- ✓ [Merca,2006] Osvaldo Mercado Torrico Sistema de Información para el Seguimiento y Control del Registro de Bu ques.

ANEXOS

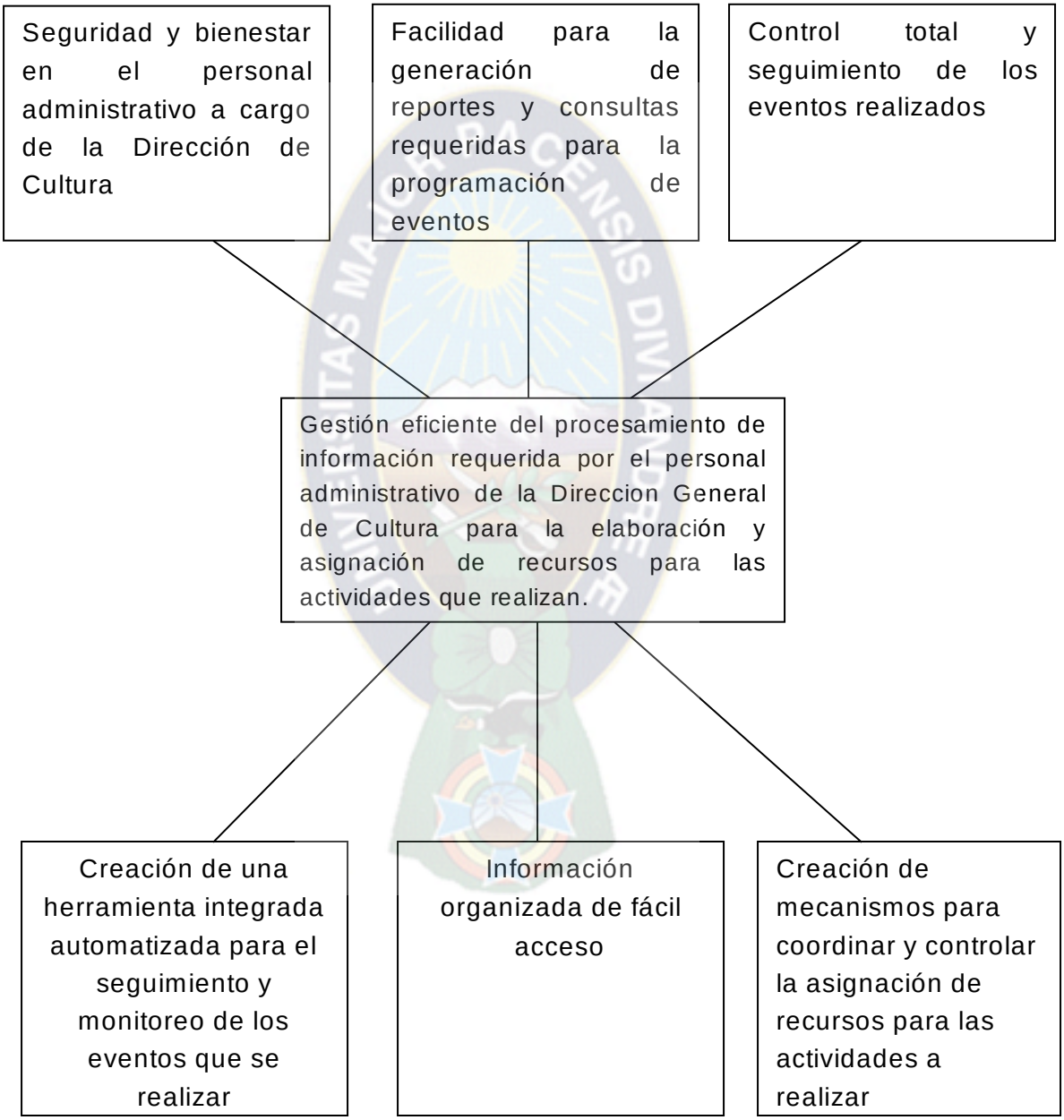
ANEXO 1

ARBOL DE PROBLEMAS



ANEXO 2

ARBOL DE OBJETIVOS

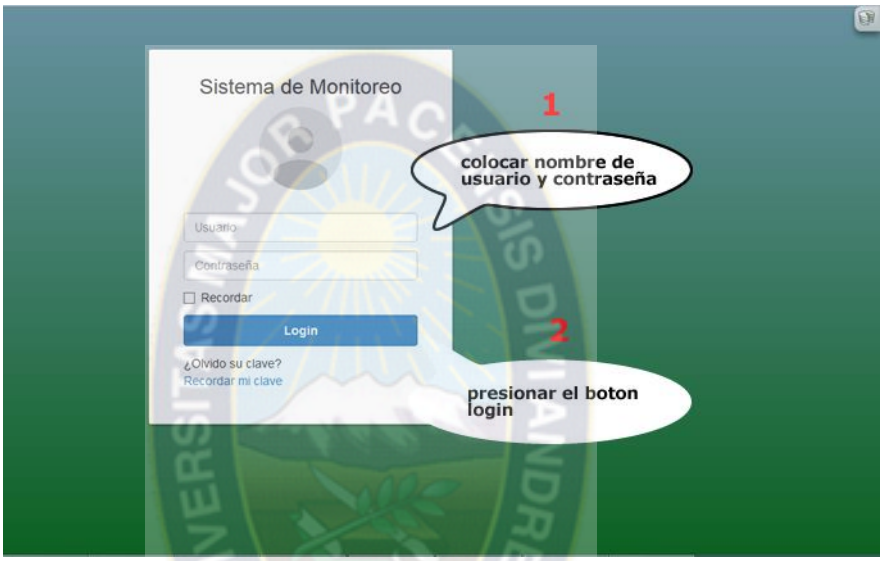


ANEXO 3

MANUAL DE USUARIO

1. Autentificación de usuarios

Para el ingreso al sistema el la persona debe contar con un usuario y una contraseña que le asignara el administrador. Y debe proceder a autentificarse.

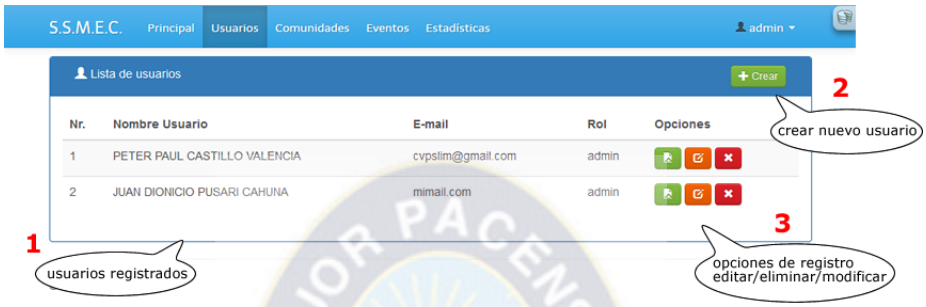


2. Pantalla de inicio sistema

Pantalla de inicio y área de trabajo una vez autentificado el usuario

3. **administración de usuarios**

listado y administración de usuarios con sus respectivas opciones de editar eliminar ver registro.



4. **Administración de comunidades**

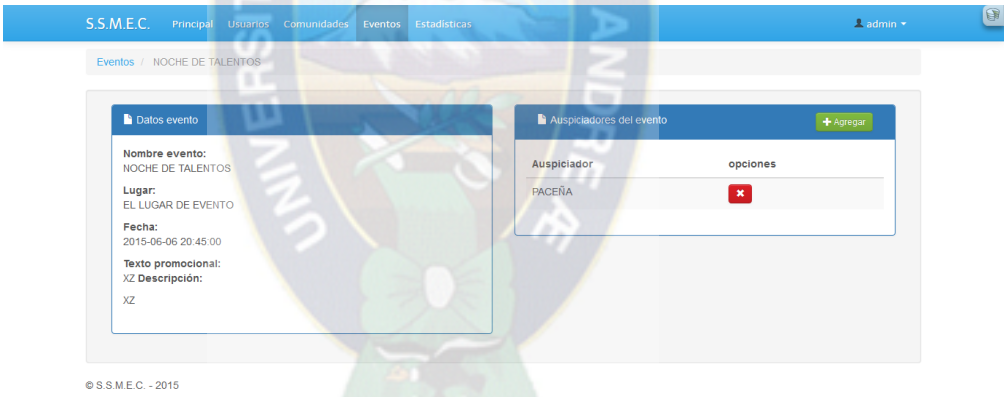
listado y administración de comunidades y sus respectivas opciones de editar eliminar ver registro. Y los formularios de registro

5. **Administración de eventos**

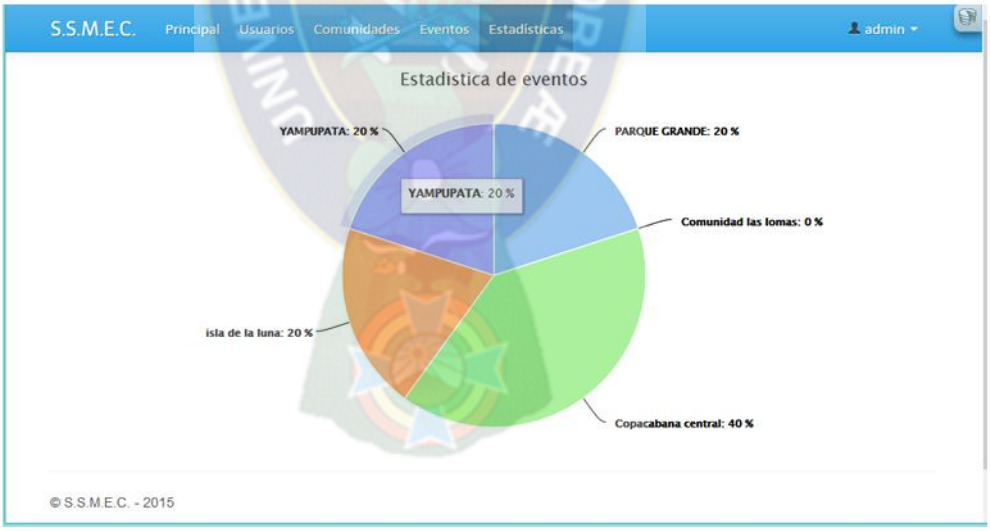
Listado de eventos y a la respectiva comunidad que corresponde con las opciones de agregar eliminar y modificar.

a) Lista de eventos

b) ficha de evento



c) **formulario de registro de eventos**

6. Estadísticas de eventos

Grafica estadística de eventos según las comunidades ejemplo de la imagen